

ESCUELA UNIVERSITARIA DE INGENIERÍA
TÉCNICA DE TELECOMUNICACIÓN
de la
UNIVERSIDAD POLITÉCNICA DE MADRID



INGENIERÍA AUDIOVISUAL Y COMUNICACIONES

TELEVISIÓN DIGITAL:

M.P.E.G.-2 y D.V.B.

Febrero 2003

Luis I. Ortiz Berenguer

PRÓLOGO A ESTA EDICIÓN ELECTRÓNICA EN 2018

Desde que en la década de los 90 apareciera la estandarización del MPEG-2 como método de codificación y método de transmisión de señal codificadas, hemos pasado una época en la que su uso ha sido prevalente, para seguir por otros momentos en que otros métodos de compresión más eficaces como el MPEG-4-parte10 (H.264/AVC) o el más reciente HEVC (H.265) le han ido sustituyendo en muchas aplicaciones.

Sin embargo, no todo ha sido sustituido. Así, la trama de transporte 'Transport Stream' sigue siendo usada como soporte para llevar estas nuevas señales comprimidas. Tampoco el DVD ha terminado de desaparecer y aún existen videoclubs y bibliotecas con fondos de películas en este soporte que usa MPEG-2. Ciertas aplicaciones de difusión siguen estando regidas por normativas que exigen el uso de MPEG-2. Y algunos fabricantes siguen recurriendo a esta señal como tecnología claramente estable y bien conocida. En el campo del audio comprimido aún se mantienen sistemas que usan difusión con audio MPEG-1/2 (p.e. DAB) y todos los soportes en MP3 (MPEG-1/2 Layer III).

Los televisores actuales, que hace tiempo que incorporan decodificadores AVC (H.264) y recientemente HEVC (H.265), mantienen intacta su capacidad de decodificar MPEG-2 (H.262) por razones de compatibilidad.

Los sistemas de difusión han ido pasando de las versiones DVB-X iniciales que se tratan en este libro a las versiones DVB-X2, aunque no todas han sido desplegadas totalmente en sustitución de las iniciales aún existentes.

Es por todo esto que merece la pena dejar acceso abierto a este libro, que puede permitir a los estudiantes y profesionales enterarse de estos aspectos aún no totalmente obsoletos.

Además la sencillez tanto de MPEG-2 como del DVB inicial, facilita la comprensión de estos temas y permite adentrarse con buena base en el estudio de los métodos más actuales, que por cierto siguen la misma estrategia básica de aquellos.

El Autor

Madrid, Septiembre de 2018.

PRÓLOGO a la edición de 2003

(ISBN: 84-95227-32-0)

Este libro responde a la necesidad de que la formación de ingenieros vaya pareja a la implementación a gran escala de sistemas de difusión de televisión digital. Esta ha sido la intención desde que en 1997 apareciera la primera edición del libro 'TV digital:MPEG-2'. Posteriormente, en 1999, el libro fue ampliado con un segundo capítulo dedicado a las normas de difusión DVB, dando lugar al nombre definitivo del libro. Ahora, esta tercera edición incluye, además de algunas correcciones que los lectores me han ido haciendo llegar y que agradezco, la ampliación de algunos datos que eran propuestas y que se han ido consolidando en estos años. También he modificado la redacción de las aplicaciones de MPEG-2 pues los últimos años han supuesto una definitiva implantación de esta señal en campos en los que antes eran sólo tentativas. En cuanto a los sistemas de difusión, todos ellos han sido desarrollados como estándares y se implementan en los distintos países de la Comunidad Europea. En todos los casos, la señal banda base de transmisión digital es conforme a MPEG-2. En España existe difusión en los tres soportes que se explican: satélite, cable y terrestre. En este caso de la televisión terrestre (TDT), su uso en España es normativo desde el año 2000, existiendo varios canales reservados para múltiplex digitales de las entidades habilitadas.

El Autor

Madrid, Febrero 2003

Sobre el autor:

El autor es Doctor Ingeniero de Telecomunicación por la E.T.S.I.T. de la Universidad Politécnica de Madrid. Desde 1986 a 1990 trabajó como Ingeniero de Sistemas de Vídeo en las empresas Vitelsa y Coratel para clientes institucionales, profesionales y 'broadcasters'; y como Jefe de Proyectos de megafonía y comunicaciones en la U.T.E. Abengoa-Sintel para los contratos con Expo92. Desde 1991 trabaja como Profesor de Escuela Universitaria en la E.U.I.T.Telecomunicación de la U.P.M. siendo Profesor Titular desde 1995. Actualmente imparte docencia en las asignaturas 'Ingeniería de Vídeo', 'Nuevas Tecnologías en Vídeo y Televisión', 'Televisión Digital' y 'Comunicaciones en Audio y Vídeo'.

El autor recibe correo electrónico en **'lortiz@diac.upm.es'**
Agradecerá cualquier comentario e indicación de erratas que ayuden a mejorar la calidad del libro y su misión formadora. También serán atendidas las consultas relacionadas con el tema del libro.

PRÓLOGO a la edición de 1999

Este libro responde a la necesidad de que la formación de ingenieros vaya pareja a la implementación a gran escala de sistemas de difusión de televisión digital. Todos ellos han sido ya desarrollados como estándares y empiezan a implantarse en mayor o menor grado en los distintos paises de la Comunidad Europea. En el caso de la televisión digital terrestre, su uso en España no es normativo hasta el año 2000. Las próximas ediciones del libro incluirán otros aspectos ya estandarizados y otros que se encuentran en fase de estandarización.

El Autor

Madrid, Febrero 1999

Sobre el autor:

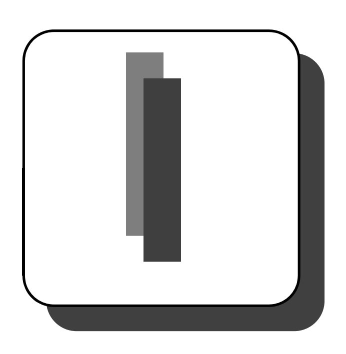
El autor es Ingeniero de Telecomunicación por la E.T.S.I.T. de la Universidad Politécnica de Madrid. Desde 1986 a 1990 trabajó como Ingeniero de Sistemas de Vídeo en las empresas Vitelsa y Coratel para clientes institucionales, profesionales y 'broadcasters'; y como Jefe de Proyectos de megafonía y comunicaciones en la U.T.E. Abengoa-Sintel para los contratos con Expo92. Desde 1991 trabaja como Profesor de Escuela Universitaria en la E.U.I.T.Telecomunicación de la U.P.M. siendo Profesor Titular desde 1995. Actualmente imparte docencia en las asignaturas 'Ingeniería de Vídeo' y 'Nuevas Tecnologías en Vídeo y Televisión'.

El autor recibe correo electrónico en
'lortiz@diac.upm.es'
Cualquier comentario, indicación de erratas o consultas serán atendidas

ÍNDICE DE MATERIAS

CAPÍTULO I: MPEG-2

CAPÍTULO II: D.V.B.



TELEVISIÓN DIGITAL: MPEG-2

INTRODUCCIÓN

En este texto se va a presentar el fundamento teórico así como las tendencias de una de las filosofías de procesado digital para compresión más importantes en el campo de vídeo y audio actual. Tal es la importancia de esta filosofía, que después de haberse utilizado y depurado extensamente durante los últimos años, ha dado lugar a una estandarización internacional de los detalles más importantes para asegurar la existencia de una metodología común para compresión, que asegure el abaratamiento de costes y la compatibilidad de señales. Nos encontramos sin duda ante uno de los hitos importantes en el desarrollo de la aplicación de las tecnologías digitales a la solución de ciertas problemáticas en el mundo de la comunicación audiovisual. De hecho, se trata de la señal banda base para la televisión digital.

La estandarización y MPEG-2

El Motion Pictures Expert Group (MPEG) es un grupo de trabajo común de la ISO (International Standards Organitation) y la IEC (International Electrotechnical Committe), concretamente el ISO/IEC JTC1 SC29/WG11 ("Joint Technical Commission 1, SubCommission 29/Working Group 11"). Este grupo, fundado en 1988 desarrolló inicialmente la recomendación ISO/IEC-11172, conocida como MPEG-1 y posteriormente ha desarrollado la recomendación ITU-H.262 ó ISO/IEC-13818 conocida como MPEG-2. Ambas recomendaciones dirigidas a la

codificación de información con fines muy concretos en el primer caso (MPEG-1) y con fines más generales en el segundo (MPEG-2).

Como ya se ha comentado al inicio, existen otros estándares, con propuestas similares para compresión, anteriores y para aplicaciones concretas. Ejemplos son el JPEG para imágenes fijas, el usado para transmisión digital de vídeo en enlaces de 34Mbps, para compresión de audio (MUSICAM), y otros.

¿Qué pretende MPEG-2?

Los estándares MPEG contemplan la compresión y descompresión (recuperación) de imágenes en movimiento y de sonido, y la formación de una estructura multiplexada de datos en forma de señal digital serie que incluya tanto el vídeo y el audio comprimidos como los datos de cualquier otro servicio auxiliar añadido. Además, contempla la sincronización adecuada durante la reproducción del vídeo, el audio y los datos entre sí. Para ello, lo que realmente estandariza es la decodificación y los datos que deben existir en la señal, sin entrar a definir los métodos concretos de codificación.

El estándar MPEG-2 intenta servir para una amplio número de aplicaciones como medios de grabación digitales, difusión de televisión y comunicaciones. Concretamente, este estándar MPEG-2 es muy flexible ya que no es un método o norma única sino que engloba un conjunto de éstas, de entre las que se elige la más adecuada en cada aplicación. El método general es el mismo, pero hay diferencias en los valores asignados a los parámetros y en las características.

¿Qué señales están involucradas?

El proceso de codificación MPEG-2 da lugar a una señal digital serie cuya interpretación no es directa ni a veces evidente, pero cuyo procesado adecuado en el decodificador da lugar a la recuperación de las señales de vídeo y audio originales, a la vez que también se pueden recuperar una serie de señales de datos auxiliares.

Esta señal digital no debe ser nunca confundida con la señal digital serie propia de equipos de vídeo sin compresión (SDI 270/360Mbps según CCIR656 ó SMPTE 259M) o de audio sin compresión (AES/EBU).

Las señales originales que usa el codificador MPEG-2 son señales en banda base tanto para vídeo como para audio, estando la de vídeo en componentes (Y, P_b,P_r) y pudiendo ser de definición estándar o de alta definición.

El uso de señales banda base analógicas exige la existencia de conversores A/D en el codificador MPEG-2, lo que encarece el equipo, por lo que generalmente se considera que la señal banda base disponible ya está en forma digital, concretamente AES/EBU para audio y vídeo digital componentes (muestreo según ITU-R601 y sus revisiones para SDTV y la correspondiente para HDTV).

Aplicaciones de la señal MPEG-2

Las características operacionales de MPEG2 afectan directamente a la manera en que probablemente el estándar es usado en varias aplicaciones. Por tanto, y a pesar de que es un estándar de amplia aceptación, MPEG2 no es la respuesta universal a la manipulación de vídeo digital.

MPEG2 beneficiará sin duda a las aplicaciones de gran escala con estructuras punto-

multipunto. En otros casos el ahorro de las soluciones MPEG2 será relativo pero el usar una señal que cumple con un estándar hará que sea empleada aunque solo se explote una fracción de las posibilidades de dicho estándar. Pero cada vez es más usado en aplicaciones punto-punto e incluso en entornos de estudio de televisión, llegando a existir interfaces de conexión específica de señal comprimida (p.e. SDTI y ASI).

A continuación se irán analizando una serie de áreas de aplicación de vídeo digital, indicándose las ventajas y desventajas de introducir el uso de MPEG2 en cada aplicación.

1.-Grandes sistemas broadcast: difusión multicanal a gran público

Las grandes aplicaciones broadcast, como difusión directa por satélite, televisión por cable y televisión terrestre digitales ya están estandarizadas en base a señal MPEG-2. Inicialmente los decodificadores son relativamente caros comparados con los circuitos analógicos similares, lo cual es especialmente relevante en estas aplicaciones de consumo a gran escala. Sin embargo la mayoría del coste de los decodificadores se debe a los componentes digitales, y estos continúan abaratándose más rápidamente que los analógicos, estrechándose cada vez más la diferencia. Estas aplicaciones están estandarizadas dentro del conjunto de normas DVB (‘Digital Video Broadcasting’).

También aparecen aplicaciones de difusión catalogadas como MDVS (‘Multipoint Video Distribution System’) estandarizadas por DVB, que operando en bandas de 27 ó 40 GHz, difunden señal MPEG-2.

2.-Vídeo bajo demanda y almacenamiento de vídeo interactivo

Los servidores de vídeo para "VOD (Video On Demand)" deben almacenar grandes cantidades de vídeo digital, para que pueda haber un acceso rápido y su posterior transmisión. Los ahorros en coste de almacenamiento usando compresión MPEG2 son muy llamativos, y aunque el acceso aleatorio (elección de un cuadro concreto) es algo más complicado que en el

caso de material sin comprimir, es posible. En este campo sigue la competencia con otros métodos de compresión

3.-Transporte por redes de telecomunicación digitales

El vídeo comprimido con MPEG2 puede distribuirse sobre muchos y diferentes sistemas de transmisión digital actuales. Considerando que cada vano individual sea de calidad razonable, la propiedad de regeneración de la transmisión digital previene cualquier degradación de la señal que no sea la introducida por la propia compresión. Esto disminuirá el coste y aumentará la flexibilidad con la que el vídeo puede ser transportado. Esto introduce a las compañías de telecomunicación en el negocio de la difusión y transporte de la televisión digital.

Existen estándares DVB (publicados por la ETSI 'European Telecommunications Standard Institute') para la interconexión de señales MPEG2 a circuitos PDH ('Plesiochronous Digital Hierarchy') y SDH ('Synchronous Digital Hierarchy') que constituyen la base de los grandes sistemas de telecomunicación.

3.1.-Transporte por redes de paquetes (con pérdida potencial de paquetes).

También dentro del ámbito de las compañías de telecomunicación, una red de paquetes puede aceptar salvas de paquetes mientras que la fuente de vídeo apenas fluctúe, abriendo la posibilidad de un calidad mejor para un régimen binario medio dado. Sin embargo, las redes de paquetes tienen el riesgo de perder paquetes cuando ocurre una congestión, y es difícil preservar la integridad del vídeo comprimido ante tal pérdida de paquetes. Actualmente las redes ATM permiten la transmisión fiable de vídeo comprimido haciendo uso de la capa de adaptación AAL-1 tal y como se especifica en varios estándares al respecto. En otros tipos de redes de conmutación de paquetes pero sin 'QoS' ('Quality of Service') asegurado, la escalabilidad proporcionaría una medida de protección, dado que se dividen los datos comprimidos en dos trenes de datos: un tren de datos de alta prioridad (que proporciona una imagen útil), y en un tren de datos de baja prioridad (que junto al anterior da una imagen mejorada), pero que puede

prescindirse de él, al perderlo durante dichos períodos de congestión. Incluir dicha escalabilidad incrementa el coste hasta el punto de preferir evitar su uso

4.-Pequeñas aplicaciones broadcast

Aplicaciones tales como difusión entre negocios de suscripción (o usuarios) limitada y difusión educacional, tienen un número de decodificadores por cada codificador mucho más pequeño que en grandes aplicaciones broadcast, por lo que son mucho más sensibles al coste del decodificador. Muchos sistemas son cerrados en el sentido de que todo el equipamiento es suministrado y mantenido por una única entidad, reduciéndose algo de la utilidad de la estandarización. Sin embargo, los beneficios potenciales en cuanto a reducción de precios a largo plazo obtenidos por compartir componentes con aplicaciones de consumo y la facilidad de interconexión con otros sistemas MPEG2 favorecerán la penetración de MPEG2 en estos sistemas.

5.-Distribución ENG

Dentro de esta actividad ("Electronic News Gathering") ha cobrando especial interés el uso de radioenlaces terrestres así como enlaces por satélite, inicialmente con modulaciones analógicas (FM). La modulación digital sin compresión exige un ancho de banda superior al disponible en estos canales. Sin embargo, la compresión permite la transmisión de una señal vídeo con calidad excelente en dichos canales.

En este caso puede haber sólo un decodificador por cada codificador, por lo que el coste del decodificador determina la economía del sistema. La ventaja de mejores accesos a los sistemas de transmisión (p.e.satélites) durante los momentos de máximo uso, y los menores costes de transmisión, motivan fuertemente el uso de compresión digital. Sin embargo, la portabilidad y la resistencia del equipo codificador son críticas, por lo que la introducción de MPEG2 en esta aplicación va paralela a la existencia de sistemas codificadores altamente

integrados competitivos con esquemas de compresión menos complejos aunque sean menos eficientes. Esta situación ya está resuelta e incluso se aprueba un estándar para DSNG ('Digital Satellite News Gathering') basado en señal MPEG-2.

6.-Multimedia y vídeo para ordenador

Existen diversos algoritmos de compresión de vídeo para uso en ordenadores, muchos de ellos en base a archivos AVI, lo que dificulta la identificación. Incluso algunos algoritmos tipo MJPEG lo hacen. Estos algoritmos proporcionan menor calidad a menor régimen binario (o menor complejidad) que MPEG-2 al trabajar cuadro a cuadro (sólo 'intraframe'). También está introducido el uso de MPEG1 por su afinidad directa con las posibilidades del soporte CD. La posible mayor calidad visual de MPEG2 es atractiva, pero requiere unos medios de mayor régimen binario y más potencia de procesado y decodificación. Los actuales ordenadores ya lo tienen y se dispone de diversas tarjetas decodificadoras. La codificación MPEG2 en este entorno es la última en incorporarse por problemas de coste y tamaño físico, aunque existen codificadores no en tiempo real por software e incluso codificadores hardware en modo ‘intraframe’.

La limitación de la lectura en soportes CD está alrededor de 1.5Mbps (datos útiles) que es la que aprovecha MPEG-1. La progresiva introducción de la familia de discos DVD ("Digital Video Disc" o "Digital Versatile Disc") junto con interfases y buses de mayor régimen binario, han mejorado las aplicaciones multimedia basadas en ordenador de propósito general y codificación MPEG-2.

7.-Consumo y entretenimiento

La implantación del DVD como soporte digital para películas de consumo supone la implantación de MPEG2 como formato de señal digital en este entorno. DVD graba películas con compresión MPEG2. El truco de su implantación es evidente, para ver películas grabadas sólo es necesario el decodificador MPEG2 que es el elemento barato de la cadena. Es la misma situación que las aplicaciones de difusión de TV digital. Otra cuestión es el abaratamiento de los grabadores DVD que incluyen un codificador MPEG2. Ese es el punto clave para su generalización como sustituto del magnetoscopio doméstico.

8.-Producción y edición en estudio

Tal y como pasa con el vídeo bajo demanda hay un gran atractivo en la reducción del coste de almacenaje de vídeo usando MPEG2, especialmente para aplicaciones de archivo, producción de noticias y edición de éstas. Las producciones y ediciones de alta calidad tienen claramente vedado el uso de compresiones elevadas, para preservar al máximo dicha calidad.

Sin embargo, salvo que el proceso se limite al uso de imágenes codificadas sin predicción (sólo 'Intraframe'), las operaciones de acceso aleatorio, inserción y conmutación son difícilmente realizables directamente sobre la señal comprimida lo que es un limitación crítica en aplicaciones de edición.

Además, debe tenerse cuidado en las operaciones de producción que involucran múltiples generaciones de manipulación de la señal. En formatos sin comprimir no sufren degradación, pero si se maneja material MPEG-2, especialmente cuando se comprime a niveles típicamente pensados para transmisión (p.e. 3-9 Mbps), tiende a mostrar rápidamente degradación al repetir los procesos de compresión/descompresión.

Para solucionar los problemas de alta compresión, permitir el trabajo con señales 4:2:2 y permitir el uso de codificación sólo-'Intracuadro' sin mucha pérdida de calidad, MPEG-2 incluye un perfil de estudio (Perfil 4:2:2) que admite regímenes binarios comprimidos de hasta

50Mbps que está teniendo una gran aceptación y ya se usa incluso en algunos formatos de magnetoscopios digitales y en servidores con grabación en disco duro.

9.-HDTV

MPEG2 soporta HDTV, y de hecho el estándar HDTV americano ("Grand Alliance", ATSC) incluye una codificación MPEG2. En Europa, estándar Eureka-256 para HDTV por satélite, aunque no usa MPEG2 propiamente, es muy parecido. Los estándares DVB ya están pensados para soportar HDTV con señal MPEG2. Sin embargo, los problemas económicos y de producción generales de HDTV retardan el despliegue: el coste de los decodificadores de alta definición, el coste de los monitores HDTV con alta resolución, y la disponibilidad de programación. La implantación de pantallas planas es un punto de referencia para la proliferación de demanda de HDTV por parte de los usuarios.

El uso de la señal HDTV comprimida en aplicaciones de producción de vídeo y de cine digital es cada vez más extendido, existiendo formatos de grabación de HDTV comprimida. MPEG-2 establece varias opciones para comprimir señal con imagen activa de 1920x1080.

10.-VCR's: Magnetoscopios digitales domésticos

MPEG2 puede disminuir el coste de un VCR en comparación con un equipo de vídeo digital sin comprimir dado que los regímenes binarios requeridos son mucho menores. Sin embargo un VCR económico (para consumo), requiere un codificador barato. La codificación MPEG2 a éste nivel de precio sufre la competencia de otros formatos para la grabación doméstica como el DV.

11.-Comunicaciones en dos direcciones (bidireccional)

Las vídeoconferencias o los enlaces en directo con retorno son casos típicos de comunicaciones bidireccionales de vídeo. En estos casos, el retardo que sufre la señal que se codifica cuando se usan imágenes tipo B puede convertir a MPEG-2 en inútil para la comunicación en dos sentidos.

MPEG-2 sin cuadros tipo B puede ser utilizado, pero con poca ventaja sobre el estándar H.320 existente para vídeoconferencia en dos sentidos a regímenes binarios por debajo de 1 Mbps. Es más, H.320 añade varios estándares adicionales útiles exclusivos de vídeoconferencias. Dichas posibilidades no están disponibles en MPEG-2. Por otro lado, MPEG-2 puede ofrecer mejor calidad visual a mayores regímenes binarios, dado su posibilidad de soportar mayores resoluciones que el formato común intermedio de H.320 (352 pixels horizontal por 288 pixels vertical).

SISTEMA MPEG-2

La definición del sistema dentro de los documentos de estandarización, hace referencia a los detalles para conseguir que una señal tan compleja como MPEG-2 pueda usarse en aplicaciones reales en las que los equipos que la generan y los que la usan son distintos y están enlazados por algún medio de comunicación.

Como ya es conocido, los sistemas de señal digital pueden organizar los datos de una manera directa y fácilmente relacionada con los datos originales (caso de las señales digitales serie banda base: SDI, AES/EBU) o de maneras más complejas y aparentemente caprichosas.

En ese segundo caso, deben preveerse elementos que permitan identificar y reordenar dichos datos para la recuperación de las señales originales. Además debe tenerse en cuenta la problemática derivada de que una señal digital incluya datos de diversas señales originales que tenían una relación temporal intrínseca (p.e. imagen y voz), lo que en general corresponde a sincronización de señales.

Por tanto, podemos hacer una división respecto al tipo de datos en base a su contenido:

-Datos del sistema (o **capa del sistema**): Conjunto de bits encargados de suministrar información sobre temporización, detalles del método de compresión usado, método usado para la multiplexación del vídeo, audio y datos auxiliares; e información para reproducción sincronizada.

-Datos de señal (o **capa de compresión**): Conjunto de bits que llevan la información útil de imagen, sonido y otros, comprimidos, además de ciertos parámetros del método de compresión usado.

Estas dos capas o estos dos tipos de datos no están claramente separados sino intercalados.

Desde otro punto de vista, a la hora de la definición del sistema, aparecen tres aspectos importantes que configuran tres fases de organización de datos:

*__Elementary Stream__ (E.S.): Datos pertenecientes a la capa de compresión, correspondientes a cada una de las señales que componen un único programa. Por ejemplo la señal de vídeo comprimida MPEG-2 (ISO/IEC 13818-2) de un programa. También forma un E.S., la señal de audio comprimida (según ISO/IEC 13818-3) del mismo programa. El E.S. es una señal digital contínua con un régimen binario concreto. Cualquier otra señal comprimida de vídeo o audio de otros programas es también una E.S.

*__Packetized Elementary Stream__ (P.E.S.): La señal E.S. se divide en paquetes de longitud no necesariamente fija, a los que se añade su correpondiente "header" (cabecera) que identifica y sincroniza los datos de dicho paquete respecto a los otros paquetes obtenidos del mismo E.S. o de otros E.S., para lo que incorpora datos pertenecientes a la capa de sistema.

Esta estructura de paquetes es necesaria de cara a la transmisión, para minimizar el efecto de errores y para permitir la multiplexación de varios E.S., que pueden tener orígenes y regímenes binarios distintos, en una única señal digital.

*__Program Stream (P.S.) y Transport Stream (T.S.)__: Son las señales MPEG-2 completas. Consisten en la sucesión de PES correspondientes a un sólo programa o a varios programas (sólo en el caso de T.S.) multiplexados con indicación de los detalles de sincronización, de acceso condicional y otros.

La razón de la existencia de dos tipos de estructuras: "Program Stream" y "Transport Stream" se debe a que cada una articula una serie de detalles adicionales a la simple agrupación de PES que las hace más adecuada a diversos tipos de aplicaciones.

Concretamente, el criterio de selección de uno u otro se basa en si se manda un sólo

programa o varios y en si el medio de transmisión al que se introduce la señal MPEG-2 va a estar sometido a errores o no.

Si son varios programas es necesario usar el T.S.. Si es un único programa puede usarse tanto el P.S. como el T.S.

Si el medio estará sometido a errores (transmisión, grabación...), la estructura a usar es la de "Transport Stream", que es la más usada evidentemente. Si el medio no está sometido a errores (conexiones cortas o interiores de equipos electrónicos) la agrupación se realiza como "Program Stream".

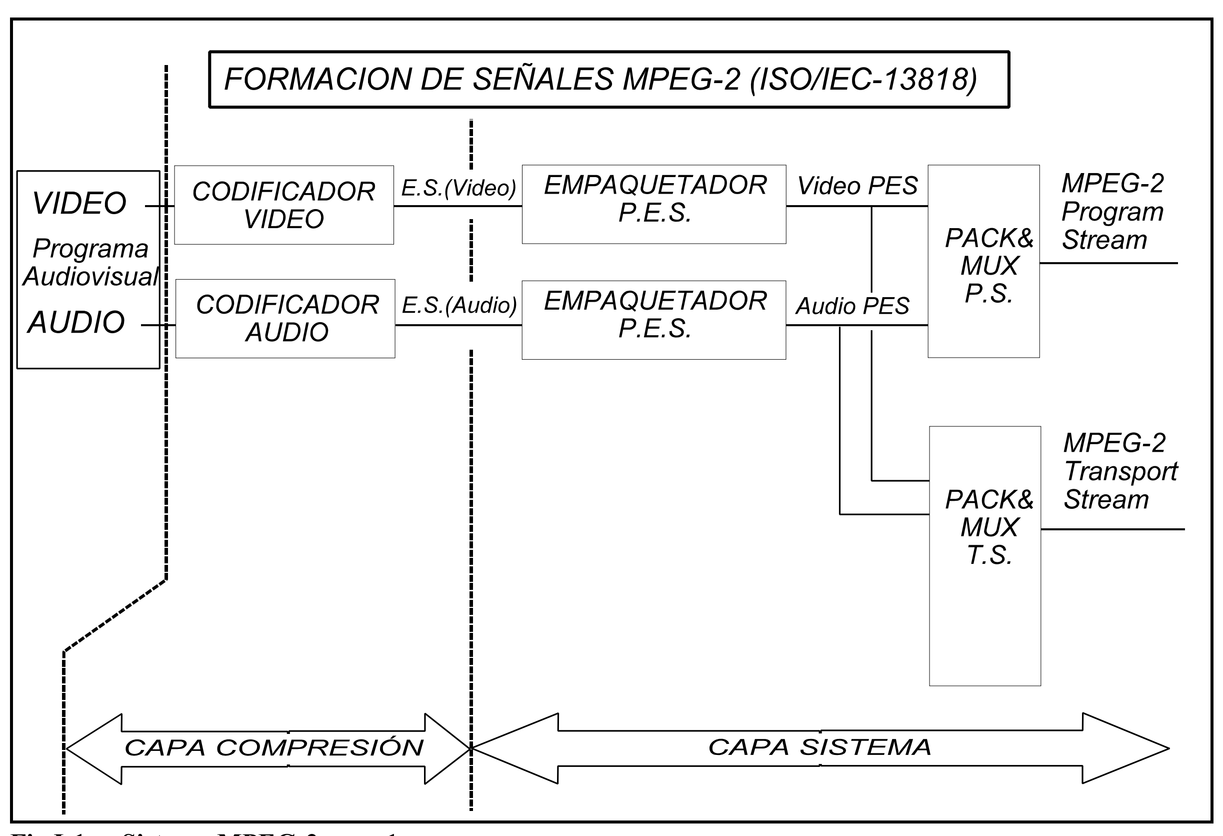


Fig.I-1.- Sistema MPEG-2 para 1 programa

FAMILIA DE CODIFICACIONES MPEG-2

La familia MPEG-2 contempla una cierta variedad de compromisos entre costo y calidad (régimen binario). A mayor calidad requerida para el proceso, mayor será el costo. Pero no siempre es necesaria la mayor calidad. Además, se contempla la aplicación de MPEG-2 tanto a señales de alta definición como de definición estándar, por lo que siempre es necesario realizar la elección. Cada nivel de calidad o prestaciones se corresponde con una serie de aspectos del procesado de la señal. Estos aspectos se agrupan en base a dos conceptos importantes: NIVEL y PERFIL. La documentación MPEG-2 define varios niveles y varios perfiles. Cada par "nivel@perfil" determina una norma concreta de la familia MPEG-2.

Niveles y Perfiles ("Level@Profile")

El **NIVEL** permite seleccionar el procesado en base a la estructura espacial de muestreo, es decir, en base al número de píxeles y líneas a codificar. El número de cuadros/sg es siempre el de la señal original antes de codificar con compresión.

NIVELES:

High	Muestreo (codificación) de 1920x1152 muestras activas
High-1440	Muestreo (codificación) de 1440x1152 muestras activas
Main	Muestreo (codificación) de 720x576 muestras activas (ver la ampliación para el perfil "4:2:2": 720x608)
Low	Muestreo (codificación) de 352x288 muestras activas

Sólo se codifican las muestras activas indicadas. Las señales originales, según la norma de muestreo digital usada (ITU-R601 y equivalentes) tendrán muestras activas y no activas. Las no activas no son codificadas por MPEG-2 y serán regeneradas con nivel de borrado en el decodificador. Las activas pueden coincidir en número con las usadas por MPEG-2 o no

coincidir. En el primer caso se recuperarán todo las muestras en la decodificación y en el segundo, las muestras activas originales no codificadas en MPEG-2 serán regeneradas con nivel de borrado por el decodificador. Para que esto sea posible, uno de los datos de sistema que se transmiten es el del formato digital original.

En los sistemas de definición estándar (SDTV) de 525 líneas y en los de alta definición (HDTV) de 1125 líneas o menos, el número de líneas codificadas es inferior al indicado en la tabla de niveles.

Parece evidente que existe una relación directa entre el tipo de señal original y el nivel a elegir. Así, una señal HDTV usará los niveles "High" o "High-1440" según su frecuencia de muestreo original en banda base. Una señal SDTV de calidad usará el nivel "Main" y el nivel "Low" se destina a señales SDTV de calidad comparable a VHS para aplicaciones de menor calidad.

Si se quiere tener un funcionamiento con escalabilidad, es necesario definir dentro de cada nivel, al menos dos niveles de calidad distintos, denominados generalmente "capa de mejora" y "capa base". Esta clasificación de "capas" es independiente de la ya comentada.

Evidentemente, el nivel usado da lugar a un régimen binario en general distinto.

De cara a la construcción de codificadores, no es necesario que todos ellos sean capaces de codificar a todos los niveles sino que existirán codificadores específicos para cada nivel. En este sentido, es importante saber que un codificador MPEG-2 que funcione a un nivel, también puede codificar a niveles inferiores.

El **PERFIL** establece un juego de "herramientas" y posibilidades en el procesado, introduciendo limitaciones en cuanto a versatilidad y escalabilidad del método de compresión-transmisión-descompresión.

Mientras que los niveles marcan claramente su capacidad para cada tipo de señal

original, no es así de evidente la selección de perfiles. Por ejemplo, el perfil "Main" es adecuado tanto para señales de definición estándar como para HDTV. Por tanto, la selección del perfil deberá realizarse en base a otros criterios.

PERFILES:

Simple	No se usan imágenes tipo B. El muestreo es 4:2:0.
Main	Se usan imágenes tipo I,P y B. El muestreo es 4:2:0.
SNR scalable	Se usan imágenes tipo I,P y B. El muestreo es 4:2:0. La transmisión es escalable en SNR, pudiéndose recibir imagen completa con menos bits/muestra (menor SNR) en caso de recepción defectuosa.
Spatially scalable	Se usan imágenes tipo I,P y B. El muestreo es 4:2:0. La transmisión es escalable en resolución espacial (número de muestras), pudiéndose recibir imagen completa con menos pixeles en caso de recepción defectuosa. Se definen dos capas: nivel base y nivel mejorado.
High	Se usan imágenes tipo I,P y B. El muestreo puede ser 4:2:0 ó 4:2:2. La transmisión es escalable tanto en SNR como espacialmente.
4:2:2	Se usan imágenes tipo I,P y B. El muestreo es 4:2:2 (aunque puede reducirse a 4:2:0). La transmisión no es escalable. Sólo utilizable con el nivel "main", Incrementando además el número de píxeles y líneas activas respecto a dicho nivel. Concretamente usa 720 muestras activas y 608 líneas/cuadro en sistemas SDTV 625/50 que tienen 720x576 de imagen activa; y 1920x1088 en sistemas HDTV que tienen 1920x1080 de imagen activa.

La selección de un método MPEG-2 exige siempre determinar un nivel y un perfil. Existen en este momento 4 niveles y 6 perfiles, por lo que podrían existir 24 posibles métodos básicos de MPEG-2. Sin embargo, no todas las posibles combinaciones nivel@perfil están consideradas, y sólo algunas pocas de las consideradas están teniendo uso actualmente.

En vídeo de definición estándar, la combinación más usada es nivel "Main" con perfil

"Main" ("Main Level@Main Profile" ó "ML@MP"). Sin embargo, para aplicaciones de estudios de producción, en donde la compresión es una fuente clara de degradación, la combinación MPEG-2 que se está dispuesto a aceptar es "Main Level@422 Profile". Ya se aceptan en estudios, para ciertas aplicaciones, otras compresiones parecidas pero que no son MPEG-2.

Regimen binario de una señal MPEG-2

Existen tres aspectos interesantes al respecto del régimen binario de una señal comprimida:

1-Régimen binario variable o fijo.

2-Régimen binario ajustable según grado de compresión y calidad

3-Régimen binario máximo

Vamos a analizarlos.

1.-Régimen binario variable o fijo.

Los métodos de compresión usados en las familias MPEG-2 incluyen varios procesos que hacen que dos imágenes con el mismo número de pixeles (p.e.720x576) y la misma duración (40ms), den lugar a distinto número de bits tras la codificación. Esto supone que cada 40ms salen un número de bits distintos y por tanto el régimen binario (Mbits/sg) varía de una imagen a otra, no es fijo.

Esto puede no ser un inconveniente en ciertas aplicaciones, pero no es lo normal en la mayoría de aplicaciones para las que se va a usar MPEG-2. Por tanto se decide que hay que forzar a que el régimen binario de la señal sea fijo, para lo cual se utiliza una memoria "buffer" como último paso del codificador del E.S.

La memoria se encarga de ir almacenando todos los bits que le llegan y de irlos sacando a un ritmo constante. Concretamente, si cada cuadro (cada 40ms) daba lugar a un número de bits

distinto y de la memoria siempre se saca un número de bits igual cada segundo, y por tanto también cada 40ms, tenemos que en 40ms se está enviando la información de más o de menos de una imagen. **No se puede considerar que una señal MPEG-2, cada 40ms contenga la información de un cuadro.**

Ejemplo:

Un primer cuadro genera 320000 bits, por lo que tendremos una entrada al buffer con régimen de 8Mbps. El siguiente cuadro genera 250000 bits, por lo que tendremos una entrada a 6.25Mbps. El siguiente genera 100000 bits, lo que supone 2.5Mbps...

Si se quiere la salida a 7Mbps fijo, sacaremos 280000 bits cada 40ms. Por tanto, el primer cuadro (eran 320000bits) tardará 45.7ms en salir. El segundo saldrá en sólo 35.7ms. El tercero en 14.3ms.

Nótese que en sólo 95.7ms se ha sacado la información de tres cuadros, que tardaron 120ms en formarse en la señal original. Esto indica que en realidad nos podemos quedar sin datos que sacar de la memoria salvo que la tengamos suficientemente llena antes de empezar a sacar. Esto supone trabajar forzando un cierto retardo a la señal que se codifica.

También podría haber ocurrido lo contrario, que siempre entren más bits de los que sacamos, por lo que la memoria tendría que ser inmensa.

Conclusiones:

*La primera idea es que se pierde la relación 40ms=Datos de 1 cuadro

*El número de bits que entran en la memoria variará respecto a un valor medio y el tamaño de la memoria debe dimensionarse de forma que la memoria no se agote ni se sature para el régimen binario de salida elegido

*La codificación exige un retardo inicial.

2.-Régimen binario fijo ajustable según grado de compresión y calidad

Es evidente que el grado de compresión determina un régimen binario de salida. Concretamente:

$$Relación\ de\ compresión = \frac{270\,Mbps*}{Régimen\ binario\ de\ salida}$$

*También suele usarse 166 Mbps, que es el régimen binario que corresponde a sólo la parte activa (720x576) a 8 bits y 4:2:2

Se puede caer en el error de pensar que un cierto grado de compresión asegura un cierto grado de calidad, esto no siempre es cierto.

Nótese que para un cierto régimen binario de salida y una cierta dimensión de la memoria buffer, debe quedar controlado el rango de variación del régimen binario de entrada a la memoria. Como se verá, este control afecta muy concretamente a los procesos de codificación y altera el grado de calidad de la imagen comprimida. Así, para un cierto régimen binario de salida, la calidad resultante dependerá de la imagen que se está codificando.

Ejemplo:

En un momento dado se está codificando una imagen que da lugar a un gran número de bits, pero por el estado de la memoria buffer, no se pueden introducir tantos bits en ella. El proceso de codificación se modifica para que la misma imagen dé lugar a menos bits. Esto tiene como consecuencia que la imagen queda codificada con menor calidad.

La situación anterior se evita si el régimen binario de salida es alto (menor grado de compresión). Sin embargo, no tiene sentido aumentar el régimen binario de salida si las imágenes a codificar son tales que no van a poner en un aprieto a la memoria. Es un nuevo compromiso a decidir.

3.-Régimen binario máximo

El régimen binario definitivo es por tanto un parámetro que se fija por el usuario, y el buffer y el sistema de control se encargan de mantener, modificando si es necesario la calidad de las imágenes codificadas.

La documentación de la estandarización recomienda unos ciertos tamaños de buffer para cada pareja nivel@perfil utilizable. También se han establecido los valores máximos de régimen binario definitivo que se deben obtener para cada pareja nivel@perfil.

El uso del valor máximo como régimen binario definitivo asegura la máxima calidad de imagen con el método de compresión utilizado.

Regímenes Binarios Máximos (Mbit/s)						
Nivel	**Perfil**					
	Simple	**Main**	**SNR Scl.**	**Spatially Scl.**	**High**	**4:2:2**
High	--	80	--	--	100 todas capas 80 capas medias y base 25 capa base	300 (I) 230 (IP) 175 (GOP)
High-1440	--	60	--	60 todas capas 40 capas medias y base 15 capa base	80 todas capas 60 capas medias y base 20 capa base	--
Main	15	15	15 ambas capas 10 capa base	--	20 todas capas 15 capas medias y base 4 capa base	50
Low	--	4	4 ambas capas 3 capa base	--	--	--

Notas: 1Mbit=10^6 bits

Las rayas -- indica que el par no está considerado

Se sombrean los pares más habituales.

(I): sólo intracuadro (IP):codificación con predicción pero GOP de sólo dos imágenes

(GOP): codificación con predicción usando GOP largo (p.e. 12 imágenes)

Nótese que si existen más de dos capas en la escalabilidad, la primera se denomina "base", la última "de mejora" y las demás "medias".

Normalmente se trabaja con regímenes binarios inferiores a estos máximos indicados. Existen estudios estadísticos sobre el grado de calidad en función del régimen binario para ciertos pares nivel@perfil que justifican poder trabajar con regímenes binarios algo inferiores a los máximos sin degradación visible respecto a usar dicho máximo[1]. Así, en SDTV se trabaja siempre por debajo de 9 Mbps el conjunto MP@ML usado para difusión, mientras que el 422P@ML de estudio se ajusta al máximo de 50Mbps si se comprime sólo intracuadro, y existe un formato de grabación que baja hasta 18Mbps usando IP. En HDTV, el sistema americano (ATSC) comprime MP@HL con GOP largo de 15 imágenes a tan sólo 19Mbps.

Para referirse al par ‘profile@level’ se usan abreviaturas estandarizadas que son:

"Profiles"	Abreviatura	"Levels"	Abreviatura
Simple	SP	Low	LL
Main	MP	Main	ML
SNR scalable	SNR	High-1440	H14
Spatially scalable	Spatial	High	HL
High	HP		

Otro aspecto interesante que el lector puede consultar en la documentación es la compatibilidad de decodificadores de cierto par nivel@perfil para decodificar señales de otros pares. Esta compatibilidad es limitada. Interesa notar que todos los decodificadores MPEG-2 son capaces de decodificar señales MPEG-1 (ISO/IEC 11172)

[1]Nakasu,E. et al. *"A statistical analysis of MPEG-2 picture quality for television broadcasting"* SMPTE Journal. Vol.105, n.11, Noviembre 1996.

CODIFICACIÓN MPEG-2 de VÍDEO: CAPA DE COMPRESIÓN

Imagen fuente e Imagen codificada

Para poder entender bien las señales comprimidas debe irse teniendo en cuenta que la información contenida en ellas no es directamente asociable a la imagen como normalmente la entendemos. En el capítulo anterior ya se han tenido pruebas de ello y en este se seguirán teniendo. Las denominaciones de imágenes usadas aquí no son universales sino que se establecen para mejor entender las explicaciones.

La misión de un sistema de compresión es realizar la codificación de una imagen de manera alternativa a la codificación de "forma de onda" a la que estamos más acostumbrados.

La codificación MPEG-2 parte de una imagen que se ha codificado según la recomendación ITU-R601. Esta es una codificación de "forma de onda" pues codifica los niveles de señal de cada pixel de la imagen. Constituye lo que podemos denominar una "**imagen original**".

Estrictamente hablando, una señal codificada según ITU-R601 sólo puede ser usada en una codificación MPEG-2 a nivel "main"(720x576). El resto de niveles corresponden a imágenes originales muestreadas con otras frecuencias aunque con estructura ortogonal en todos los casos. Teniendo esto en cuenta, debe permitirse que los textos suelan simplificarse con la licencia de hacer siempre referencia a señales ITU-R601 como señales originales.

Así pues, una imagen original siempre será una imagen en componentes digitalizada con estructura ortogonal, con una cantidad de **píxeles activos** NxM, con codificación de las tres componentes Y,Cb,Cr según la estructura frecuencial seleccionada (a:b:c ; p.e. 4:2:2), y con todos y cada uno de los datos codificados con Q bits.

El primer paso que realiza MPEG-2 es seleccionar el muestreo y número de píxeles activos con que trabajará en base al nivel y a las posibilidades del perfil:

Por nivel:	
High	Muestreo de 1920x1152 muestras activas
High-1440	Muestreo de 1440x1152 muestras activas
Main	Muestreo de 720x576 muestras activas
Low	Muestreo de 352x288 muestras activas
Por posibilidades del perfil:	
Muestreo 4:2:0	
Muestreo 4:2:2	

La señal original modificada para ceñirse a estos valores, se denomina **"imagen fuente"**.

Las posibilidades por nivel son autoexplicativas, si bien no es necesario que la imagen original coincida con el nivel a usar pues se puede usar un nivel inferior o superior sin más que remuestrear con diezmado o con interpolación según el caso.

Las posibilidades por perfil merecen más explicación. El caso de 4:2:2 ya es conocido al ser el usado en ITU-R601. El caso 4:2:0 sin embargo puede ser más equívoco, y además es el más usado dentro de MPEG-2.

La estructura conocida como 4:2:0 supone que el muestreo de las componentes de diferencia de color se realiza a mitad de frecuencia (mitad de puntos) tanto en el muestreo horizontal como en el vertical. Esta es una gran diferencia frente a los muestreos típicos de señales componentes (4:2:2; 4:1:1; 4:4:4;...) en los que siempre se muestrea el color en todas las líneas, por lo que no hay submuestreo en vertical y sólo lo hay en horizontal.

El submuestreo vertical no es inconveniente para aplicaciones de difusión, pues sistemas

tradicionales como el PAL realizan en el receptor un promediado de líneas que equivale a un submuestreo vertical, al que los espectadores están acostumbrados.

Una vez que se han tomado las muestras de la imagen original que realmente se van a codificar en base al par nivel@perfil, es decir, la **"imagen fuente"**, el codificador MPEG-2 procede a la codificación con compresión. Hasta este punto, se puede seguir asociando a cada pixel definitivo, una serie de valores de forma de onda (luminancia o luminancia+color), codificados todos ellos con el mismo número de bits.

La codificación MPEG-2 destruye la asociación anterior mediante el uso de varios procesos concatenados, de modo que la **imagen codificada** ya no es sino un conjunto de datos con un orden y significado distintos a los acostumbrados en la recomendación ITU-R601 y que en el decodificacor serán capaces, por sí solos o con ayuda de imágenes ya disponibles en el decodificador, de reconstruir la imagen fuente.

Una vez recuperada en el decodificador la imagen fuente, ésta deberá pasarse al formato más adecuado para su salida a la aplicación de destino (p.e.: ITU-R601, componentes analógicos, formato del adaptador gráfico de ordenador...).

La documentación ISO/IEC 13818-1 que define la capa de sistema de MPEG-2, suele referirse a los datos que configuran la imagen fuente y la imagen codificada como "unidad de presentación" y "unidad de acceso" respectivamente. En este texto, por razones de claridad, evitaremos esa denominación hasta llegar a los apartados de la capa de sistema.

A partir de ahora nos referiremos a la imagen fuente llamándola indistintamente "fuente", "original" o "a codificar".

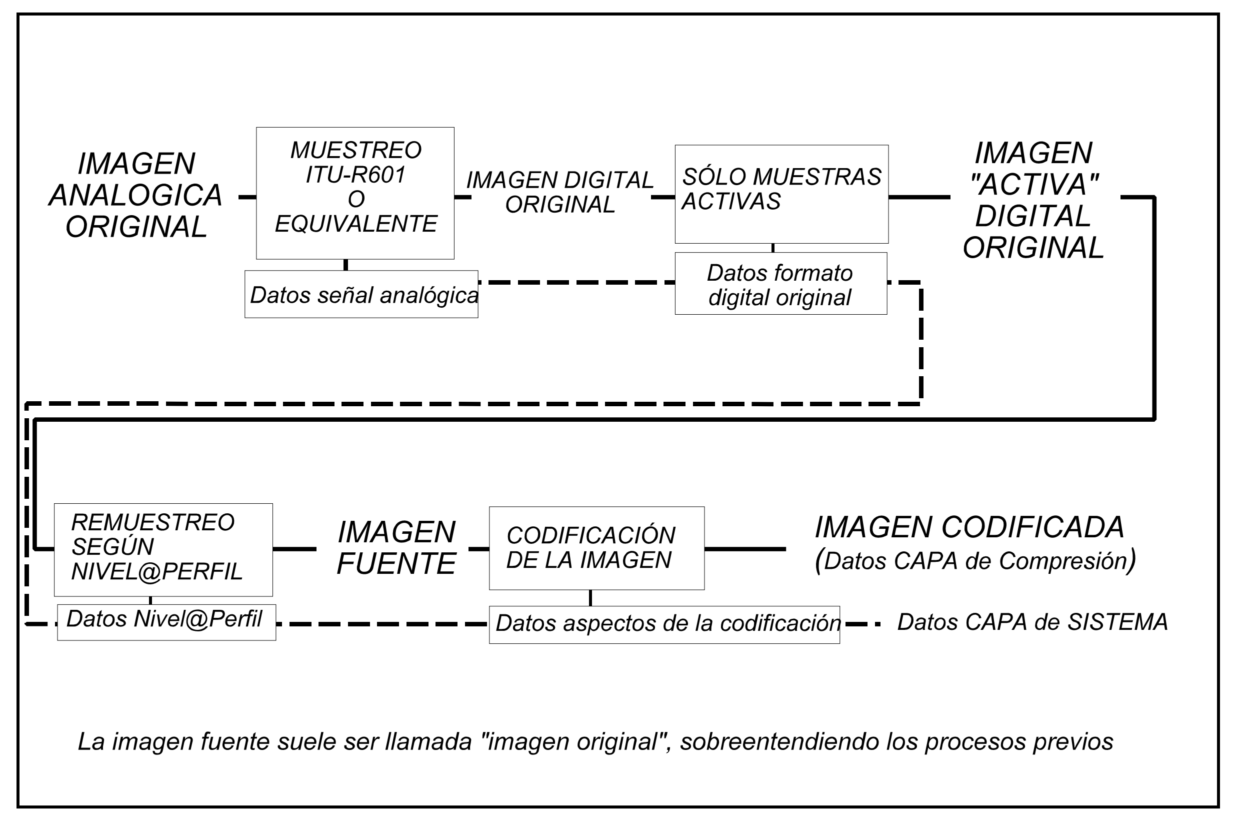


Fig.I-3.- Esquema general y terminología usada

Elementos que se codifican en MPEG-2

Ya se ha dicho que la compresión requiere la codificación de datos que pueden ser alternativos a los valores de forma de onda. Estos datos no tienen por que ser todos del mismo tipo, sino que puede tratarse de varios. Concretamente, MPEG-2 hace uso de los siguientes tipos de datos:

*Datos de imagen
*Vectores de movimiento entre imágenes
*Errores de predicción por compensación de movimiento

Estos datos se calculan a partir de la imagen a codificar y, en los dos últimos casos, también de las imágenes anteriores y/o posteriores a la que se está codificando. Esta es otra gran

diferencia frente a las codificaciones en que los datos de todas y cada una de las imágenes dependen sólo de ellas mismas.

Cada uno de estos tipos de datos se irán entendiendo en las explicaciones que siguen en este capítulo.

Proceso de codificación

Una vez que se han calculado los valores de los datos indicados (datos imagen, vectores movimiento, errores de predicción), deben codificarse. Se indican a continuación los procesos que usa MPEG-2, sin especificar detalles asociados a ellos, que ya se irán desvelando:

*Para datos de imagen y datos de errores de predicción: DCT+VLC
*Para vectores de movimiento: VLC.

Estos procesos son:

-DCT ("Discrete Cosine Transform"): La transformada discreta del coseno consiste en realizar una operación matemática a los datos originales de manera que se obtienen otros datos que se relacionan unívocamente con aquellos (operación inversa "IDCT" sin pérdidas). Los datos originales para la operación se seleccionan en bloques de NxN (típico 8x8) formando matrices. Los datos resultantes tienen las siguientes características:
-Igual número de datos y ordenación en bloque NxN.
-Datos con valores **reales** (no complejos) positivos y/o negativos.
-Significado "espectral" respecto a datos originales.
-Gran cantidad de valores cercanos a cero (muchos más que en el bloque original).

-VLC ("Variable Length Coding"): Esta codificación asigna un número distinto de bits a las palabras digitales. Así, cada dato a codificar no usa el mismo número de bits y por tanto tiende a dar lugar a un régimen binario no fijo. La asignación del número de bits a cada palabra se realiza en base al dato que se envía con esa palabra y a la probabilidad de que aparezca dicho dato.

Los datos más probables, o que más veces se dan, se codifican con palabras de pocos bits y los menos probables con palabras de más bits. Como existe un número limitado de datos que se pueden tener, existen un número limitado de palabras que enviar y se pueden elegir. De hecho, la codificación VLC supone la existencia de una tabla estandarizada que asocia a cada dato una palabra de una cierta longitud.

En el caso de MPEG-2, la codificación VLC usada se basa en el método "Huffman" con un estudio estadístico previo. Además, se realiza para la asignación de palabras una codificación "run-length" encubierta (no en proceso previo sino a la vez). Una codificación "run-length" simplifica la codificación de largas cadenas de ceros. Para ello, codifica un dato alternativo: "número de ceros consecutivos", en vez de codificar cada cero como dato independiente. Por tanto, codificar datos que sean ceros supone menor régimen binario que datos que no sean cero.

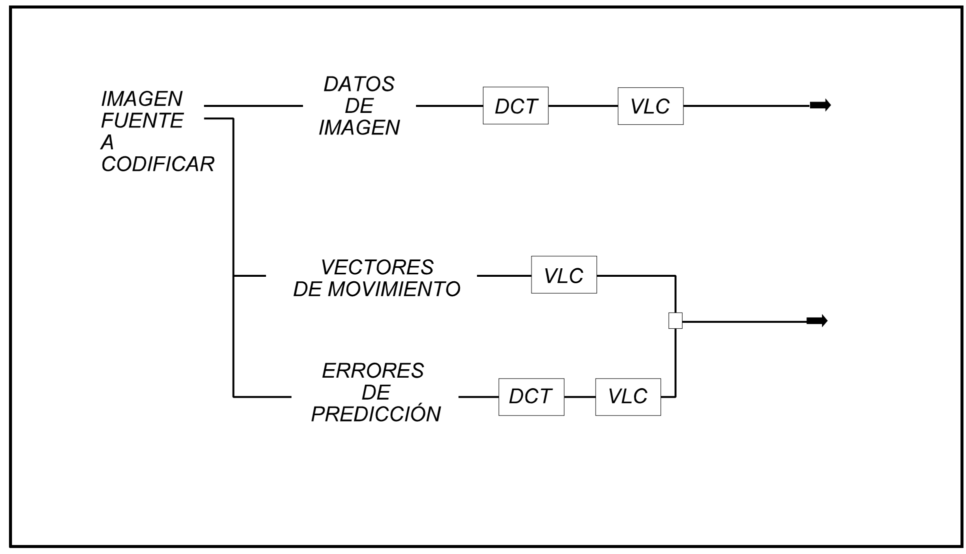


Fig.I-4.- Procesos de codificación.

Recuantificación y ponderación de la DCT

El uso exclusivo de técnicas DCT y VLC en su definición estándar, daría lugar a un sistema con un grado de compresión dependiente sólo de los datos a codificar y en definitiva de las imágenes a codificar. Esta situación puede no ser óptima en dos aspectos de la codificación MPEG-2, dando lugar a:

-Insuficiente relación de compresión

-Saturación del buffer ante ciertas imágenes complejas.

Si se quiere rebajar mucho el régimen binario de salida, se puede saturar el buffer. Para poder actuar sobre el régimen binario que entra al buffer, inicialmente fijado por las imágenes que se está codificando, se debe poder alterar alguno de los dos métodos de codificación. Como ya se ha dicho, el VLC usa tablas fijas estandarizadas, por lo que el único proceso alterable es el de la DCT.

La alteración de la DCT supone la existencia de pérdidas irrecuperables en el proceso de codificación-decodificación y por tanto una pérdida de calidad, que tiene que asumirse al decidir que el régimen binario de salida sea menor. En realidad no se altera el cálculo de la DCT sino los valores obtenidos. Los dos métodos de actuación sobre la DCT, que controlan el régimen binario que entra en el buffer son:

-Recuantificar los datos resultantes con menos bits.

-Ponderar la matriz o bloque de datos resultantes para anular más datos.

-Recuantificar los datos resultantes con menos bits: Evidentemente, si se utilizan menos bits por dato, el régimen binario disminuye. Suele decirse que este proceso simplemente supone usar un paso de cuantificación más grande para cuantificar los valores de la DCT. Esta afirmación puede resultar un poco equívoca pues al usar un paso mayor, cambia el valor del escalón cuántico en que se encuentra el valor a recuantificar. Este nuevo valor del escalón cuántico es el nuevo valor cuantificado. Veamos un

ejemplo: Los valores de la DCT, según la norma, requieren 12 bits para ser cuantificados todos sin perder ninguno (van de -2048 a +2047 tomando sólo los valores enteros). Supongamos que tenemos el valor 2000 y queremos cuantificarlo con sólo 8 bits. Es matemáticamente imposible. En realidad, el cambio del paso de cuantificación en un cierto factor exige la división, por el mismo factor, de los datos a cuantificar, para obtener la nueva cuantificación (el nuevo número del escalón de cuantificación). En el mismo ejemplo anterior, pasar de 12 a 8 bits supone incrementar el paso de cuantificación en 2^4=16 veces. Así, el dato 2000 debe dividirse por 16 (resulta 125) para cuantificarse con 8 bits.

Usar menos bits supone que algunos valores no son cuantificables y deben sustituirse por los valores más próximos, por lo que aparece un error de redondeo que no es recuperable, disminuyendo la calidad de la imagen codificada. Del ejemplo anterior, debe considerarse que todos los datos que al dividir por 16 den un resultado decimal se truncan a un único valor entero. En todos esos datos se ha cometido un error de cuantificación. Concretamente, todos los valores entre 2000 y 2015 (incluidos) al dividirse por 16 y truncar dan 125 y por tanto todos se codifican con 8 bits como el mismo valor. Puede verse que este error es equivalente a incrementar a 16 el paso de cuantificación en los datos originales.

-**Ponderar la matriz o bloque de datos resultantes para anular más datos**: Al hacer que más datos sean cero, la VLC dará menor régimen binario al buffer. Para conseguir esto, se ponderan los datos dividiéndolos por un valor de forma que se haga más pequeño el resultado. Esto, unido al paso de cuantificación, hace que los valores muy pequeños, al truncarlos, se cuantifiquen como cero. Los datos anulados no podrán recuperarse a su valor real y se perderán.

Dado que ambos procesos se realizan en una estructura circuital con el bus de datos paralelo, la ponderación es una operación que parece evidente, pero parece "problemático" disminuir el número de bits (¿Quizás quitando pistas del bus?. ¡No!). Esto se realiza

manteniendo el número de bits fisicos pero poniendo a cero algunos de ellos, lo que equivale a disminuir el rango de valores cuantificables. No es problemático sino consistente con lo que ya se ha visto sobre que todos los datos deben dividirse por un cierto factor para ser recuantificados con menos bits, pues esa división es el resultado de poner a cero ciertos bits del bus. Del ejemplo usado, si tenemos que para pasar de 12 a 8 bits (quitar 4 bits), el valor 2000 pasa a ser el valor 125, por lo que los 4 bits más significativos son cero.

Con el proceso de la división por un factor, se asegura que basten 8 bits para cuantificar un valor que inicialmente necesitaba 12 bits, pero a costa de que otros valores cercanos a este se cuantifiquen erróneamente con el mismo valor de 8 bits.

En cualquier caso, la estructura electrónica real hace innecesario el debate del número de bits del bus paralelo que entra al codificador VLC. Independientemente del número de bits físicos, lo importante es que el margen de variación de los datos previamente divididos, sea suficientemente reducido para asegurar el uso de pocos bits.

Téngase en cuenta que el definitivo garante de la disminución del régimen binario es el proceso VLC, que asigna el mismo código al valor 125 independientemente de si corresponde a un dato original 125 con 12 bits o al dato original 2000 con división por 16 (reducción a 8 bits). El decodificador recuperará el valor 125, y si la información de recuantificación y ponderación indican que el factor de división fué 1 (escalón o paso de cuantificación=1), el resultado será un dato recuperado de 125. Si indican un factor de 16 (escalón o paso de cuantificación=16), el dato recuperado será 125x16=2000 (La documentación ISO/IEC dice que la cuantificación inversa consiste en multiplicar el valor recuperado por el escalón de cuantificación).

En definitiva, tanto la disminución del número de bits como la ponderación se llevan a cabo mediante una operación de división de los valores de los datos. De hecho, de la documentación se desprende que lo que realmente se hace en la codificación es simplemente dividir cada dato por un cierto valor que queda definido por dos parámetros: **uno específico** para cada dato de la matriz DCT 8x8 y **otro general** que afecta por igual a todos los elementos de la matriz (llamado "factor de cuantificación"). Los específicos se agrupan formando una matriz 8x8

(llamada "de ponderación" en el proceso de decodificación, pero llamada "de cuantificación" en la definición de la estructura de la señal).

En resumen:

El funcionamiento real supone la división de todos los datos por unos coeficientes, en principio distintos para cada dato de la matriz DCT, que disminuyen su valor. Este valor resultante se sigue codificando con 12 bits. Sin embargo, al tomarse valores dentro de un margen menor, los bits más significativos se van a cero y pueden obviarse. Los valores obtenidos se truncan provocando un error pues varios valores DCT originales dan el mismo valor resultante. Los que den valores menor de 1 pasan a ser cero.

Los datos de la matriz de ponderación o cuantificación así como el factor de cuantificación se envían al decodificador junto con la señal, en la capa de compresión. Esta matriz supone el uso de 64x8=512 bits para cada bloque. Para poder disminuir su influencia en el régimen binario resultante, se han definido unas matrices estándares que pueden usarse por defecto y que el decodificador ya conoce para lo que basta con enviar al decodificador unas banderas e identificativos en vez de todos los datos de las matrices.

Imagenes codificadas "Intra-cuadro" e "Inter-cuadro"

Una vez indicadas las generalidades sobre la codificación, volvamos a referirnos a los datos que se pueden codificar.

Se dice que una imagen se está codificando "Intra-cuadro"("Intra-frame") si los datos que se codifican se obtienen sólo de ella. En este caso, la codificación sólo incluye "datos de imagen". Los datos de imagen son los valores de la forma de onda en cada pixel. Estos son sometidos al proceso concatenado de DCT+VLC y ya no son valores de forma de onda.

Se dice que una imagen se está codificando "Inter-cuadro"("Inter-frame") si los datos que

se codifican dependen no sólo de ella sino de otras imágenes anteriores y/o posteriores. En este caso, la codificación sólo incluye vectores de movimiento y errores de la predicción por compensación de movimiento. En el próximo apartado se concretan ciertos detalles.

Se suele hablar de "imagen intracuadro" y de "imagen intercuadro", lo cual literalmente no parece indicar lo que ocurre, y deben entenderse dichas expresiones como "imagen codificada intra/inter cuadro".

La codificación de imágenes "inter-cuadro" se corresponde con la filosofía genérica de "codificación predictiva", por lo que también suelen llamarse "imágenes predictivas" (aunque realmente signifique "imagen codificada predictivamente").

Codificación Predictiva (Imágenes Inter-cuadro)

En la codificación predictiva, para codificar una imagen NxM, se parte de otra imagen NxM previamente existente en el codificador (imagen base para entendernos) y se le aplica a dicha imagen base un proceso concreto que nos permita obtener una nueva imagen NxM que sea prácticamente igual que la imagen que queremos codificar. Este proceso se denomina "algoritmo de predicción". Entre la imagen a codificar y la obtenida mediante el algoritmo de predicción existe una diferencia (error de predicción) que se evalúa pixel a pixel y da lugar a una "imagen NxM de error". Los datos codificados son los de la imagen error y los del algoritmo, la imagen base se supone existente también en el decodificador.

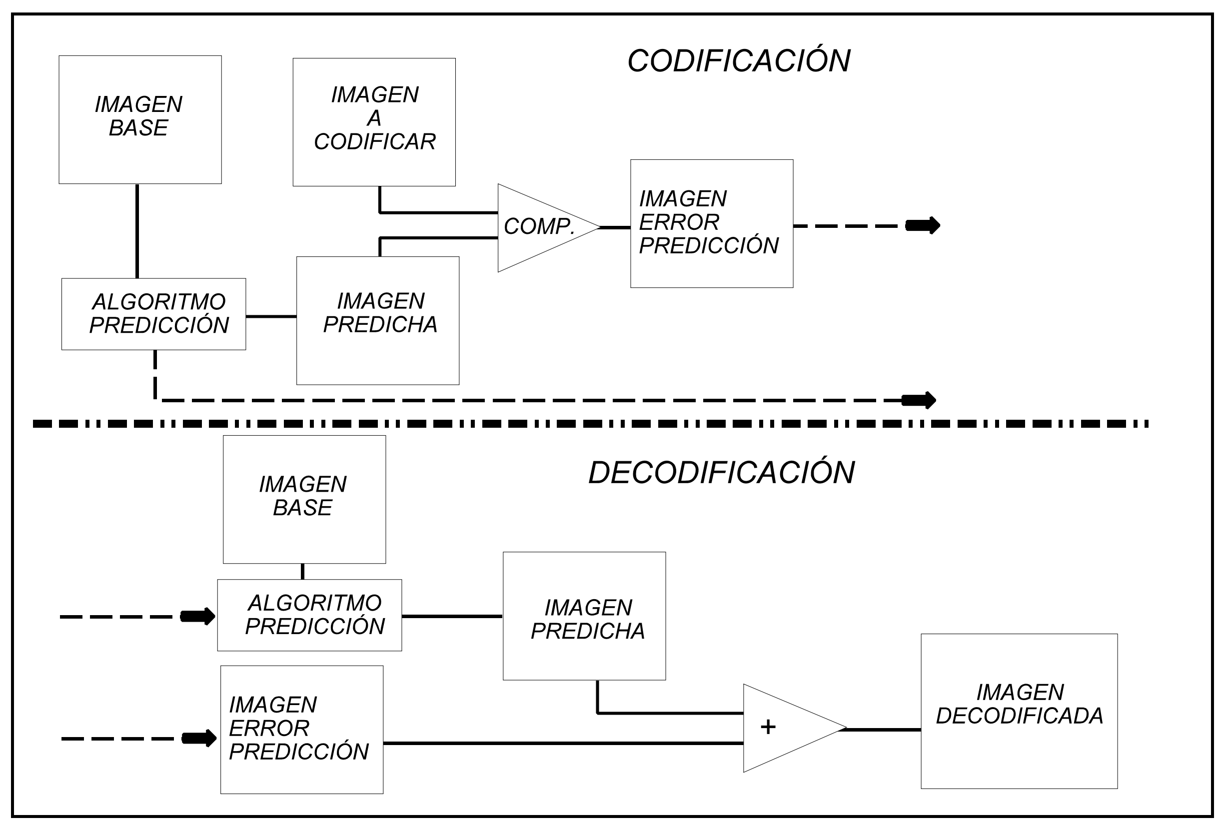


Fig.I-5.- Esquema general de codificación predictiva

Margen dinámico del error y capacidad de compresión

Si el algoritmo de predicción es suficientemente bueno, el error será pequeño en todos los píxeles: el margen dinámico del error será menor que el margen dinámico de la imagen original a codificar. La imagen de error puede ser codificada a continuación, como si se tratase de una imagen intracuadro, con menor número de bits que si se codificase intracuadro la imagen original.

Sin embargo, en general no es suficiente con trasmitir al decodificador los datos de la imagen de error, también hay que transmitirle los datos del algoritmo predictor. Aún así, el número de bits totales debe ser menor que si no se usara predicción, dando lugar a compresión. Si en algún caso no fuera así (p.e. codificar una imagen muy distinta a la imagen base), el codificador debería parar el proceso y usar una codificación intracuadro de la imagen original.

Estos datos recibidos (algoritmo e imagen error), junto con la imagen base ya decodificada anteriormente, permiten al decodificador reconstruir la imagen actual.

El principal inconveniente de la codificación predictiva está en el hecho de que cualquier error de transmisión que afecte a la recuperación en el decodificador de una imagen que sirva de base para otra imagen dará lugar a una nueva imagen con error. Es decir, los errores se propagan por todas las imágenes que se van obteniendo de forma predictiva.

La solución es que periódicamente se envíen imágenes codificadas intracuadro para que no se siga acumulando el error.

La codificación predictiva frente a la codificación intracuadro sólo tiene sentido cuando se puede obtener una reducción en los datos a mandar y por tanto una compresión.

Algoritmos de predicción

Este es sin duda el punto clave de los procesos predictivos. Para ciertos tipos de señales es posible determinar un algoritmo fijo o prefijado para realizar una predicción con un error resultante bajo. Esto permitiría evitar mandar al decodificador la información del algoritmo pues se conoce y es fijo y se implementa en el diseño del equipo.

Un ejemplo es la predicción diferencial en que el algoritmo consiste en usar la muestra anterior como predicción de la nueva. Resulta útil para codificar valores medios de zonas de una imagen respecto al valor medio de una zona anexa.

En otros casos, aunque el algoritmo no sea fijo, sí se puede conseguir que siga una cierta ecuación (p.e. polinomio en z^{-1}) y que se pueda adaptar los coeficientes de dicha ecuación para conseguir el error mínimo para el tipo de señal que se codifica en cada momento. Estos procesos se denominan "Codificaciones Adaptativas", los errores conseguidos son mínimos y sólo se tiene que transmitir al decodificador los coeficientes del algoritmo predictor.

Un detalle importante es que los algoritmos de predicción no son realmente "adivinos" sino que en la mayoría de los casos siempre comprueban el error cometido, es decir, siempre tienen presente la señal que debe obtenerse.

El algoritmo de predicción depende del mecanismo que gobierna la variación de la señal a codificar. En el caso de imágenes de vídeo, el mecanismo que gobierna el cambio de imágenes es el movimiento, ya sea el de los objetos que están dentro del encuadre como del movimiento de la cámara (panorámicas, picados, zooms, travelings...). Una imagen se diferencia de otras en que los objetos que había han cambiado de posición, en que algunos objetos que habían ya no están (se han salido) y en que otros que no había han aparecido o entrado.

En esta situación, es difícil encontrar un algoritmo simple para predicción.

Predicción por compensación de movimiento

La solución más usada para realizar predicción con imágenes vídeo es la denominada "compensación de movimiento".

Se trata de empezar por analizar el movimiento que existe entre la imagen a codificar y la imagen que va a servir de base de la predicción. En esta labor no hay nada de "adivinación" ni de estudios estadísticos sobre el movimiento, sino simple cálculo aproximado del movimiento existente en cada caso.

Este cálculo puede hacerse por varios métodos y es aproximado pues no analiza el movimiento de cada pixel sino el movimiento de trozos de imágen. El resultado es un conjunto de valores denominados "vectores de movimiento" que indican el sentido de movimiento de cada trozo de la imagen base para obtener aproximadamente la imagen a codificar.

Posteriormente, a cada trozo de la imagen base se le aplican los vectores de movimiento calculados antes y se obtiene una nueva imagen que se dice que está "compensada en movimiento" y que se parece a la imagen a codificar salvo en:

- unos errores en la colocación exacta de los trozos movidos.
- que no aparecen los objetos que existen en la imagen a codificar pero no existían en la imagen base.

La diferencia entre la imagen compensada en movimiento y la imagen a codificar es la "imagen error".

Los datos que se transmiten son los vectores de movimiento (análogos a los coeficientes del algoritmo de predicción) y la imagen error, que se codificará como una intracuadro, sufriendo el proceso DCT+VLC (aunque con una pequeña diferencia respecto a la codificación de una imagen intracuadro 'fetén', que se indicará en el próximo capítulo).

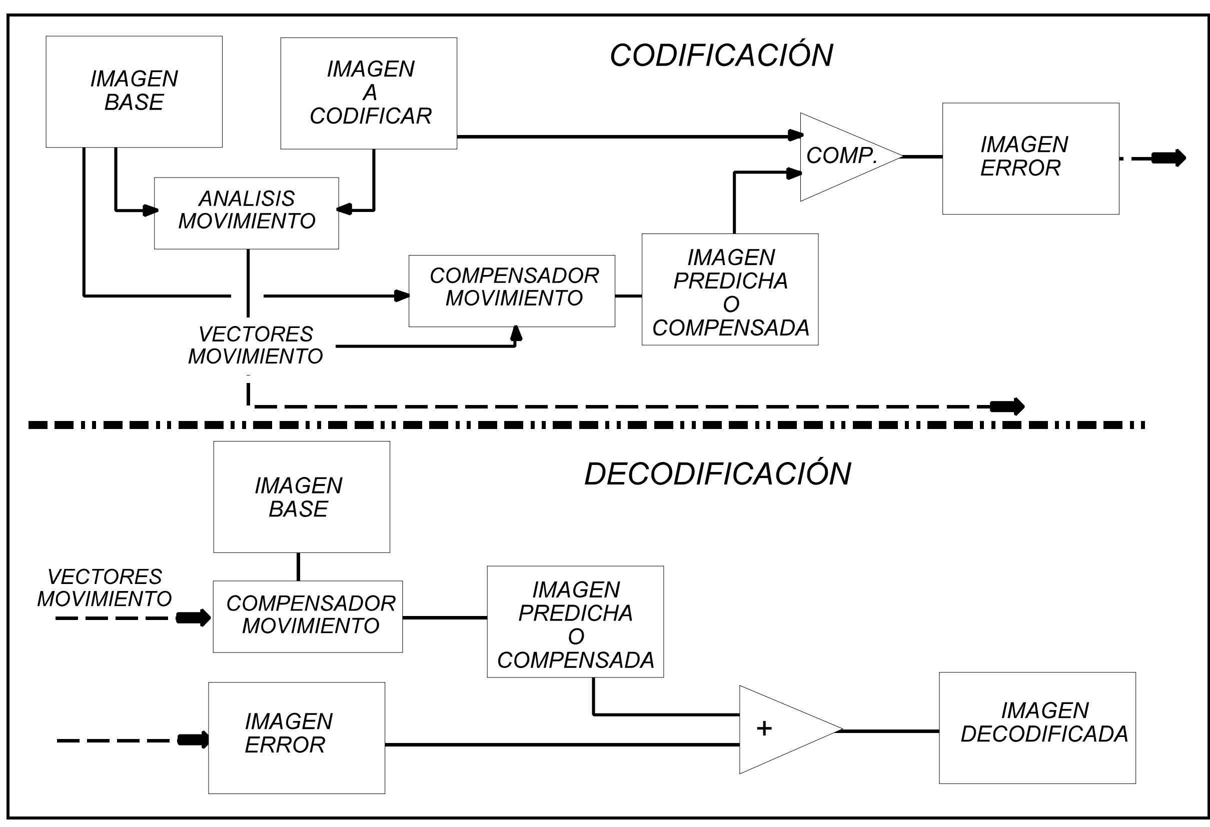


Fig.I-6.- Codificación predictiva con compensación de movimiento

CODIFICACIÓN de VÍDEO: ESTRUCTURA DE DATOS

Como se ha visto hasta ahora, los procesos de codificación se realizan sobre datos que

pueden ser de imagen original, de imagen error o vectores de movimiento. Tanto el cálculo de estos datos como su posterior codificación con procesos DCT y VLC exigen la definición de una estructuración.

Subcapas de compresión: "capas" MPEG-2

La siguiente subdivisión hace referencia a cómo se estructuran los datos de vídeo comprimidos (capa de compresión) y su explicación es larga. En resumen, se tienen las siguientes capas (en realidad subcapas, pero se admite la licencia), de menor a mayor complejidad:

-Capa de Bloque
-Capa de Macrobloque
-Capa de bandas ("slice")
-Capa de imagen
-Capa de grupo de imágenes (GOP: "Group Of Pictures")
-Capa de secuencia

Cada capa se constituye como conjunto de elementos de la capa inferior más una serie de bits de identificación de la capa.

Capa de Secuencia

La capa de secuencia se constituye como una serie de bits existentes entre un código de comienzo de secuencia ("sequence header") y un código de final de secuencia, y que contienen

la información comprimida correspondiente a varios grupos de imágenes (GOP´s).

Dentro de cada secuencia se pueden identificar varios GOP's. Como ya se sabe, MPEG-2 no transmite exclusivamente en base a codificación "intra-frame", por lo que no siempre se puede reconstruir una imagen independientemente de las otras. Algunas de las imágenes se codifican y deben reconstruirse en función de otras imágenes (codificación predictiva o diferencial), por lo que siempre se deberá decodificar una cantidad mínima de imágenes, incluso cuando sólo se quiera obtener una. Este grupo mínimo de imágenes que deben decodificarse en conjunto para recuperar las originales identifica un GOP.

Capa de G.O.P. ("Group Of Pictures")

Los GOP's permiten el acceso aleatorio a imágenes concretas y vienen delimitados por imágenes que se codifican y decodifican sólo en función de sí mismas ("Intraframe coded pictures").

Un detalle importante a recordar es que de todo el tren de datos correspondiente a un GOP de N imágenes, cada imagen no contribuye con el mismo número de bits, dado que no todas se codifican igual. Cada imagen constituye la capa de imagen.

Capa de Imagen

La capa de imagen no sólo consiste en un agrupamiento de subpartes de la imagen sino en una identificación del método de codificación/decodificación de ésta. A saber, debe informarse al decodificador si los datos que se envían corresponden a una imagen codificada:

- a partir de sí misma: "Intraframe coded" (Imagen-I)
- a partir de una imagen anterior: "Predicted pictures" (Imagen-P)

- a partir de una imagen anterior y otra posterior[2]: "Bidirectionally coded" (Imagen-B)

En esta diversidad de posibilidades de codificación reside una buena parte de la potencia de reducción de régimen binario de MPEG-2. Para minimizar el error acumulado en la predicción, las imágenes tipo P sólo se obtendrán a partir de otras imágenes tipo P ó tipo I. Un error producido en una imagen se propaga a todas las imágenes del mismo GOP.

Un GOP suele definirse por dos valores que se refieren a los tipos de imágenes que lo componen. El primer valor indica el número total de imágenes del GOP y el segundo indica la distancia entre la imagen I y la primera imagen P del GOP. Por ejemplo, un GOP (15,3) tendrá la estructura 'IBBPBBPBBPBBPBB'

Capa de Bandas ("Slice Layer")

Los bits correspondientes a una imagen se subdividen a su vez en agrupamientos inferiores. Dentro del conjunto de bits de una imagen se identifican agrupamientos denominados tiras o bandas (capa de bandas,"slice layer"), que se corresponden con tiras horizontales de la imagen. Estas tiras se corresponden en la imagen con un ancho de 16 líneas en vertical y un largo variable que puede, como máximo, igualarse al ancho de la imagen. Además, varios procesos de predicción y algunos valores por defecto se reinician al comienzo de un "slice". Las bandas están ordenadas según aparecen en la imagen en el sentido de barrido normal.

Capa de Macrobloques

Los bits correspondientes a una banda se subdividen en base a Macrobloques (capa de

[2]Cuando se dice que la imagen es anterior o posterior se refiere a su posición dentro de la secuencia real de vídeo. Sin embargo, en el proceso de codificación-decodificación, ambas se tienen con anterioridad a la imagen que se está codificando. Esto supone una reordenación de cuadros y un retardo inherente en el proceso.

macrobloques). Un macrobloque contiene el tren de bits correspondiente a la información de 256 puntos de la imagen (pixeles). Estos 256 pixeles no son cualesquiera, sino que corresponden a un cuadrado de 16x16 pixeles de la imagen, es decir, cubre la información de un cuadrado de 16 líneas de imagen y 16 pixeles por línea. Dentro de una banda, los macrobloques se ordenan en el mismo orden del barrido de vídeo: de izquierda a derecha.

La estructuración de la información de los componentes de luminancia y diferencias de color se realiza en base a macrobloques. Si de cada pixel del macrobloque se codifica tanto la luminancia como las dos componentes de crominancia se dice que se tiene una codificación 4:4:4. Si sólo se trabaja con la crominancia de la mitad de los pixeles, pero de todas las líneas, se tiene muestreo 4:2:2. Sin embargo, lo más normal en MPEG-2 es que sólo se trabaje con la crominancia de la cuarta parte de los pixeles del macrobloque, despreciando la información de crominancia de la mitad de las líneas: es un muestreo denominado 4:2:0.

El orden de los datos es, primero la información de la luminancia de todo el macrobloque y luego la información de las dos diferencias de color.

Al analizar el macrobloque, también debe tenerse en cuenta que la información de cada pixel (tanto luminancia como crominancia) puede corresponder a la propia del pixel (caso de imágenes-I) o a la de la "imagen error" (caso de imágenes-P y B).

Además, los vectores de movimiento se calculan para cada macrobloque.

Capa de Bloques

La información del macrobloque está a su vez constituida por grupos de bits que llevan la información de BLOQUES. Un bloque es el conjunto de datos correspondientes a un tipo de información de los pixeles de un cuadrado de 8x8 pixeles (p.e. hay 4 bloques de luminancia en 1 macrobloque). Sobre estos bloques de 8x8 se realiza el proceso de la DCT para la codificación. Los valores obtenidos se van ordenando en una estructura más compleja según las capas que hemos ido indicando.

En cuanto a la información de color, para sistema 4:2:0 se tiene que en un macrobloque 16x16 píxeles físicos, hay información de luminancia de todos ellos, pero hay sólo 64 píxeles de los que se toma la información de color. Estos 64 píxeles dan lugar a 1 bloque de 8x8 datos para Cb y otro bloque de 8x8 datos para Cr. Se dicen que son los dos bloques de color, pero no tienen relación física con 8x8 píxeles consecutivos de imagen. En definitiva, en 4:2:0, cada macrobloque correspondiente a 16x16 píxeles, se compone de 6 bloques 8x8 de datos.

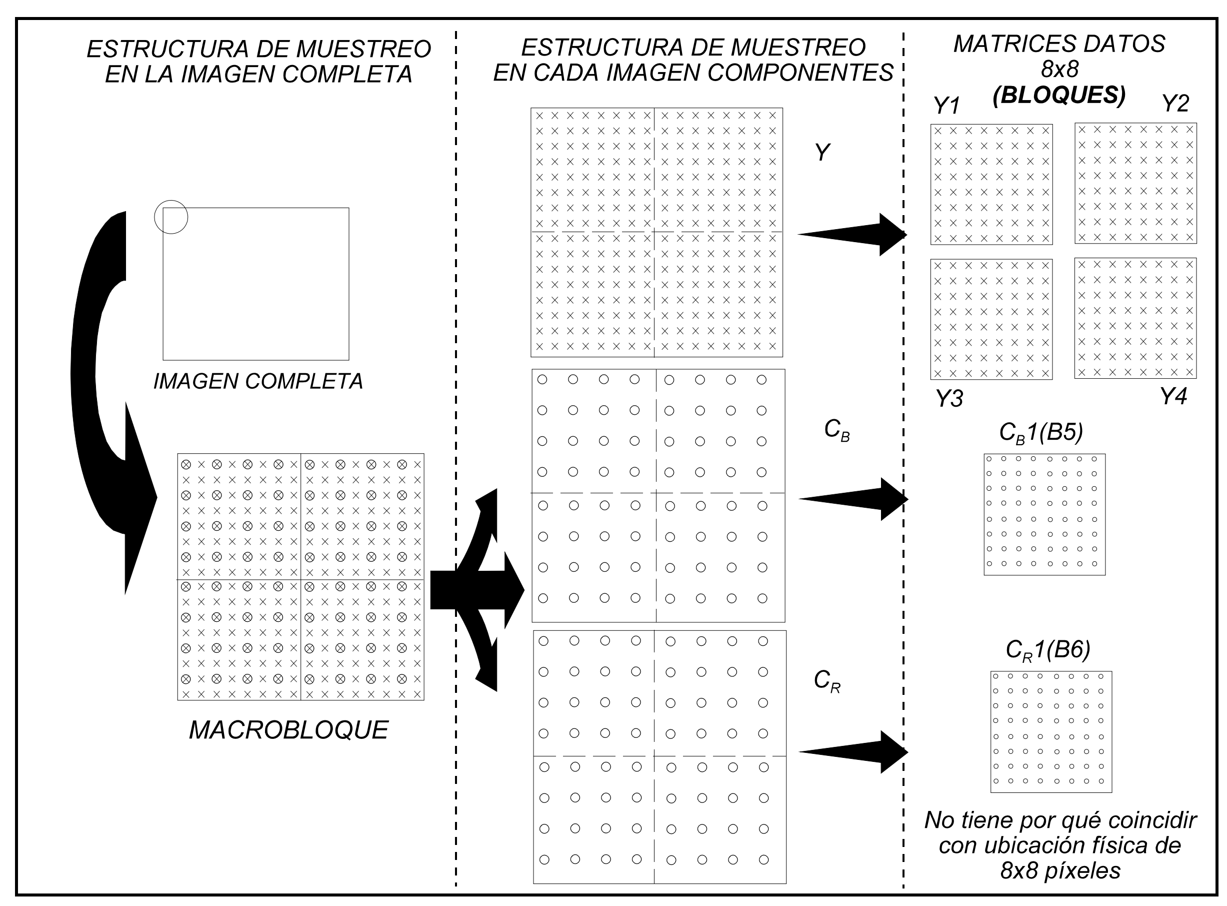


Fig.I-7.- Detalle de la estructura de bloque y macrobloque en muestreo 4:2:0

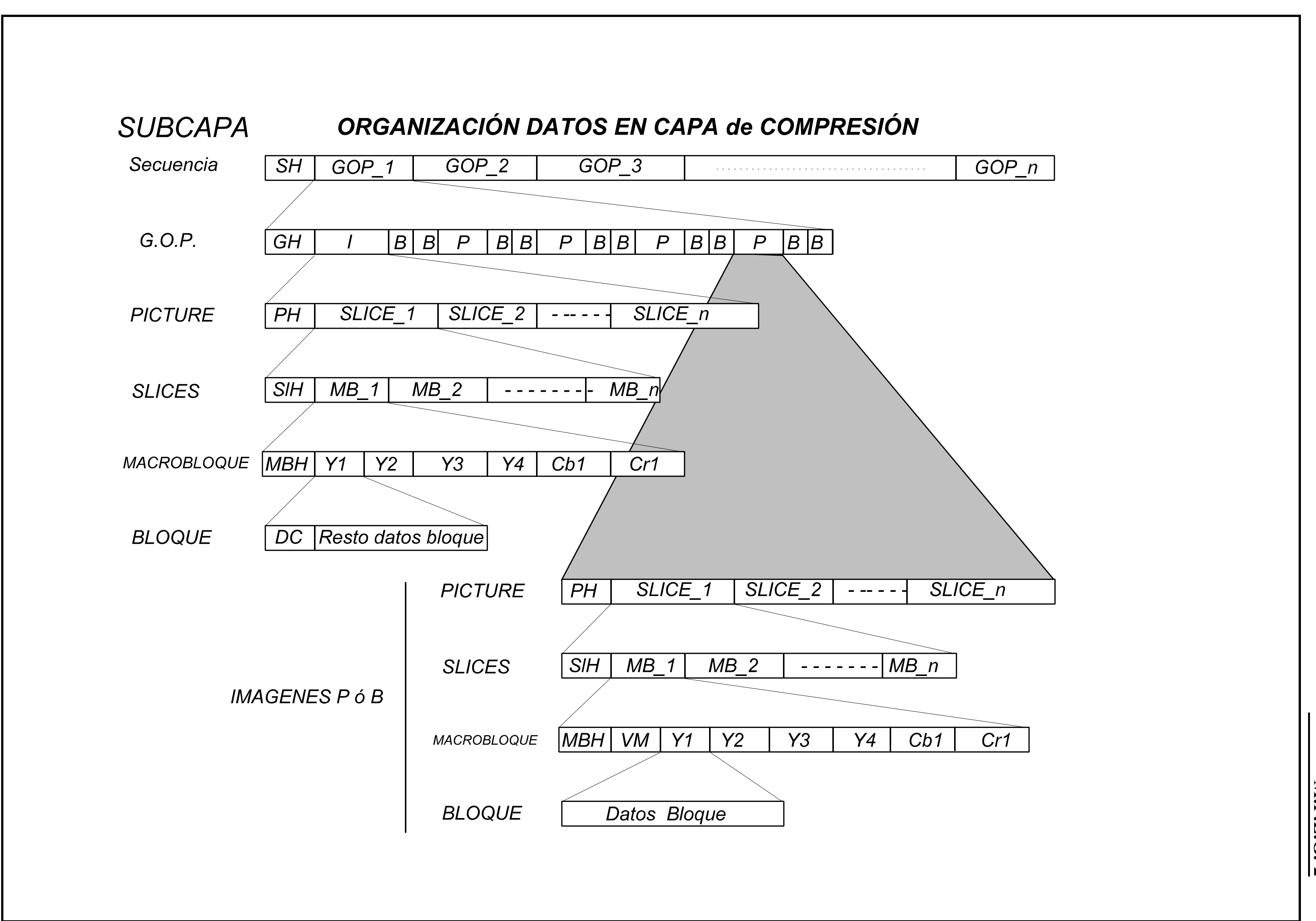


Fig.I-8.- Agrupación de los datos por capas (subcapas de la capa de compresión) para 4:2:0.

CODIFICACIÓN MPEG-2 de VÍDEO: PROCESO DE CODIFICACION

Una vez analizado el método de ordenación de los datos según una estructura en capas, se indicarán los detalles del algoritmo de codificación usado.

La imagen se digitaliza según un método clásico de codificación de forma de onda, obteniéndose los valores de las muestras de todos los píxeles. Se procede a seleccionar los datos de color teniendo en cuenta la estructura de muestreo de la crominacia que se quiera usar en la codificación MPEG-2 (p.e. 4:2:0). Este proceso puede hacerse con 8 ó 10 bits.

Codificación de imagen tipo I (Intraframe coded)

Si una imagen se va a codificar tipo I, el proceso es el siguiente:

Se van tomando los bloques de 8x8 de luminancia del primer macrobloque, en el orden del barrido de vídeo: de izquierda a derecha y de arriba a abajo, y luego los de crominancia. A cada bloque de 8x8 (64 valores) se le aplica un algoritmo de transformación DCT ("Discrete Cosine Transform") que da como resultado otros 64 valores ordenados también en matriz 8x8, pero cuyos valores ya no tienen una correspondencia directa pixel a pixel con los de la imagen original (aunque se pueden recuperar todos los pixeles originales a partir de estos 64 valores del espacio transformado).

La recomendación establece que el proceso de cálculo de DCT se realice con datos de entrada de 9 bits y valores de salida de 12 bits.

Estos 64 valores o coeficientes DCT, no tienen por qué codificarse todos con 12 bits, de hecho muchos de ellos se codificarán con menos, lo que equivale a una recuantización con el consiguiente error en los coeficientes que se van a transmitir. Cada uno de los 8x8 coeficientes

tienen una correspondencia con aspectos espectrales de la imagen y el sistema visual humano detectará los errores de forma distinta en cada coeficiente (concretamente, el ojo es menos sensible a los errores de los coeficientes de altas frecuencias que a los de baja), por lo que en la codificación se determinará con cuantos bits se recuantiza y codifica cada coeficiente. Se obtiene así la llamada "matriz de cuantificación de bloque" que es un conjunto de 8x8 valores enteros. Además, ciertos bloques pueden necesitar valores de matriz de cuantificación más altos todos que otros bloques, especialmente si la imagen varía suavemente de nivel. Esta matriz es un dato auxiliar que deberá transmitirse al decodificador junto con el resto de datos. MPEG-2 admite un cierto cambio en la matriz de cuantificación para cada bloque.

Así pues, los 64 datos (coeficientes DCT) que deben transmitirse por cada bloque, no siempre necesitan el mismo número de bits, por lo que su longitud es variable. Esto hace que sea conveniente emplear un "código de longitud variable" para la codificación. Estos códigos suelen optimizarse en la forma de códigos "variable run length" (p.e. Huffman), que se benefician además de la existencia de secuencias largas de ceros seguidos.

Concretamente, la DCT genera, en señales de imagen, gran cantidad de coeficientes nulos. Si la lectura y transmisión de los coeficientes DCT del bloque no se realiza en orden normal (por filas), sino en un orden que agrupe coeficientes nulos entre dos coeficientes no nulos, se tiene una serie de 64 datos con muchos ceros seguidos entre datos no nulos, que se puede recodificar en la forma de pares cuyos datos sean: [número de coeficientes nulos hasta el próximo coeficiente no nulo, valor de dicho coeficiente no nulo]. De esta manera,el número de datos a transmitir baja pues todo grupo de ceros se manda con un sólo dato (codificación "run-length"). Para conseguir estas secuencias largas de ceros, la lectura de los coeficientes se realiza en "zig-zag" como aparece en la figura.

Realizando el correspondiente estudio estadístico de las distintas posibilidades de pares "run-length" obtenibles de los coeficientes DCT para un bloque de imagen de 8x8 pixeles, se define el código Huffman concreto a utilizar por MPEG-2, el cual usa códigos más cortos para los pares más probables, si bien los menos probables hacen uso de códigos más largos de lo normal, pero el balance final es el de usar menos bits que si no se usaran códigos Huffman para

enviar los coeficientes DCT. El propio método de elección de códigos Huffman elimina la necesidad de añadir un identificador de comienzo y final de código.

Este proceso compuesto de "run-length" y de código Huffman para los pares se denomina simplemente "VLC (Variable Length Coding)", al hablar de la norma MPEG-2.

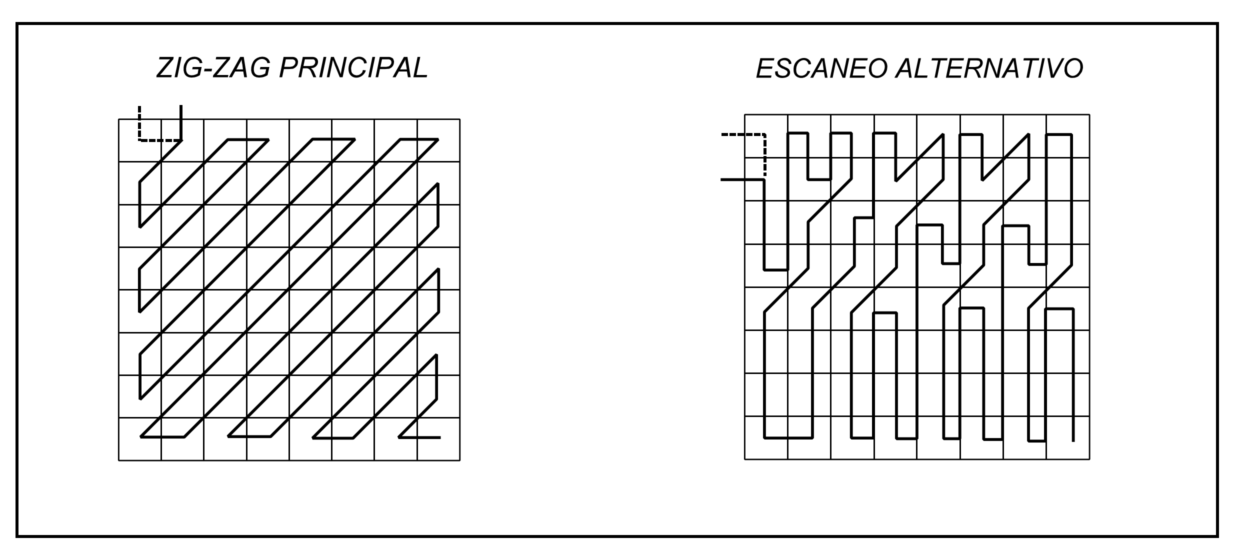


Fig.I-9.- Ordenación de datos DCT en zigzag y alternativo.

Para disminuir aún más el régimen binario, si a partir de un coeficiente de los 64, el resto es todo ceros, se envía un código de "final de bloque" en vez de los datos (ceros) restantes. El decodificador, al recibir este código, rellena de ceros los coeficientes que le resten para llegar a los 64.

Existe una excepción a lo explicado anteriormente de la codificación VLC y es el primer coeficiente DCT del bloque 8x8. Este coeficiente corresponde al valor medio (componente DC) del bloque y se cuantificará con el máximo número de bits (seleccionable entre 8 y 11 bits, aunque pueden usarse menos). Además, para la codificación de este coeficiente en los sucesivos bloques, se usa un método predictivo específico disminuyendo así su aportación al régimen binario.

Este algoritmo de predicción no es distinto para los coeficientes DC de los bloques de luminancia o de los bloques de diferencias de color, pero evidentemente existen tres procesos

de predicción simultáneos, cada uno de ellos con su predictor, para ir recuperando las tres informaciones. Los datos que se envían son los errores de predicción. La predicción se reinicia ante ciertos sucesos, para evitar acumulación de errores, ajustando los predictores a un valor preestablecido. Esto ocurre por ejemplo, al comienzo de un "slice", cuando se inicia un macrobloque no intracodificado o cuando un macrobloque no se codifica y se salta al siguiente. El predictor usado es el más sencillo y es simplemente "el valor anterior", por lo que la información de error de predicción no es sino la diferencia del valor actual y el anterior.

Los valores obtenidos de esta codificación predictiva del componente DC de la DCT de las imágenes Intra-codificadas son posteriormente codificados con un método específico compuesto de dos valores que dan como resultado códigos de longitud variable sin ser una codificación VLC como la usada para los otros coeficientes.

Codificación de imágenes tipos P y B

Si la imagen a codificar va a ser tipo P (predicción), significa que la información a codificar no es propiamente la imagen sino la información necesaria para obtener la imagen a partir de una imagen anterior conocida (imagen base). La información de predicción es de dos tipos:

-Vectores de movimiento calculados entre la imagen a codificar y la imagen base ya conocida. Los vectores de movimiento se calculan en base a macrobloques (16x16) por lo que no pueden definir a la perfección el cambio de la imagen base a la imagen a codificar. Además, sólo se analiza el movimiento en luminancia.

-Error o diferencia entre la imagen a codificar y la imagen que se obtiene a partir de la imagen base aplicando los vectores de movimiento calculados (imagen predicha).

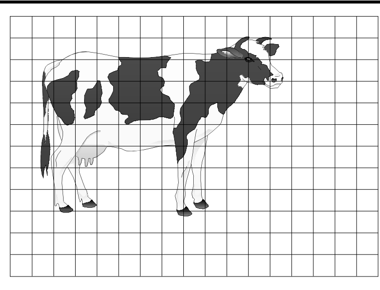
La información de los vectores de movimiento se obtiene por técnicas de detección de movimiento (p.e. "block matching") y se asignan a cada macrobloque (1620 macrobloques en

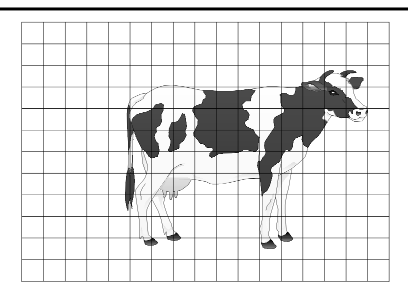
720x576).

Estos vectores de movimiento indican el sentido y tamaño del desplazamiento (componente horizontal y componente vertical) de cada macrobloque de la imagen base para obtener la imagen que se está codificando.

El decodificador, al recibir esta información, y habiendo mantenido memorizada la imagen base, la modifica en base a los vectores de movimiento (proceso llamado Compensación de Movimiento).

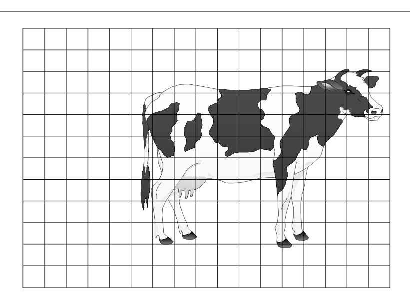
El resultado no es exactamente igual a la imagen que se quería mandar, pues los vectores de movimiento en base a macrobloques no es un sistema que represente a la perfección los procesos reales de movimiento en una imagen. Además, si la nueva imagen tiene partes de la imagen de nueva aparición o que desaparecen respecto a la imagen base, no es posible codificarlas como un movimiento de macrobloques.





Un análisis del movimiento en base a macrobloques nos indica en este caso que:

En general y aproximadamente, cada macrobloque se obtiene moviendo el macrobloque base, 4 distancias a la derecha y 1 abajo.



Como se ve, hay diferencias entre esta imagen y la que se quiere codificar. Estas diferencias son los errores que también hay que codificar.

Si se modifica el criterio de movimiento de los macrobloques pueden reducirse los errores en ciertas zonas de la imagen, pero aumentarán en otras.

Imagen aplicando sólo vectores de movimiento
Cada macrobloque mantiene su información interior intacta

Fig.I-10.- Compensación de movimiento para codificar cuadro tipo P

Por eso es imprescindible añadir la información de error entre esa imagen recuperada con compensación de movimiento y la imagen que realmente queremos transmitir. El codificador por tanto también realiza el proceso de compensación de movimiento que realizará el decodificador y obtendrá la misma imagen que obtendría el decodificador. A continuación calcula pixel a pixel la diferencia o error respecto a la imagen base. Esto da un total de 720x576 píxeles que forman la "imagen error" (luminancia más diferencias de color). Nótese que si el movimiento o cambio entre las imágenes no ha sido muy elevado, los valores de error serán pequeños y podrán codificarse con pocos bits. En caso contrario, no supone una reducción (compresión) de bits codificar la imagen como tipo P, pues la cantidad de información (vectores más errores grandes) superaría a la necesaria para codificar esa misma imagen como tipo I.

Si los errores son suficientemente pequeños, el codificador da el visto bueno a la codificación tipo Predictiva y codifica los 414720 "píxeles de error" con el mismo proceso que se usaría si fuese una imagen tipo I, es decir, transformación DCT en base a bloques 8x8, recuantificación de los valores obtenidos con posible uso de matriz de ponderación, lectura de

los valores en zig-zag y codificación combinada "run-length+VLC Huffman". En este caso, los valores de DC de cada bloque, no se codifican aparte con método predictivo.

El problema del método anterior de codificación predictiva (tipo P) se ve claramente si queremos codificar con respecto a una imagen base anterior, una imagen con un objeto que no existe en la imagen base.

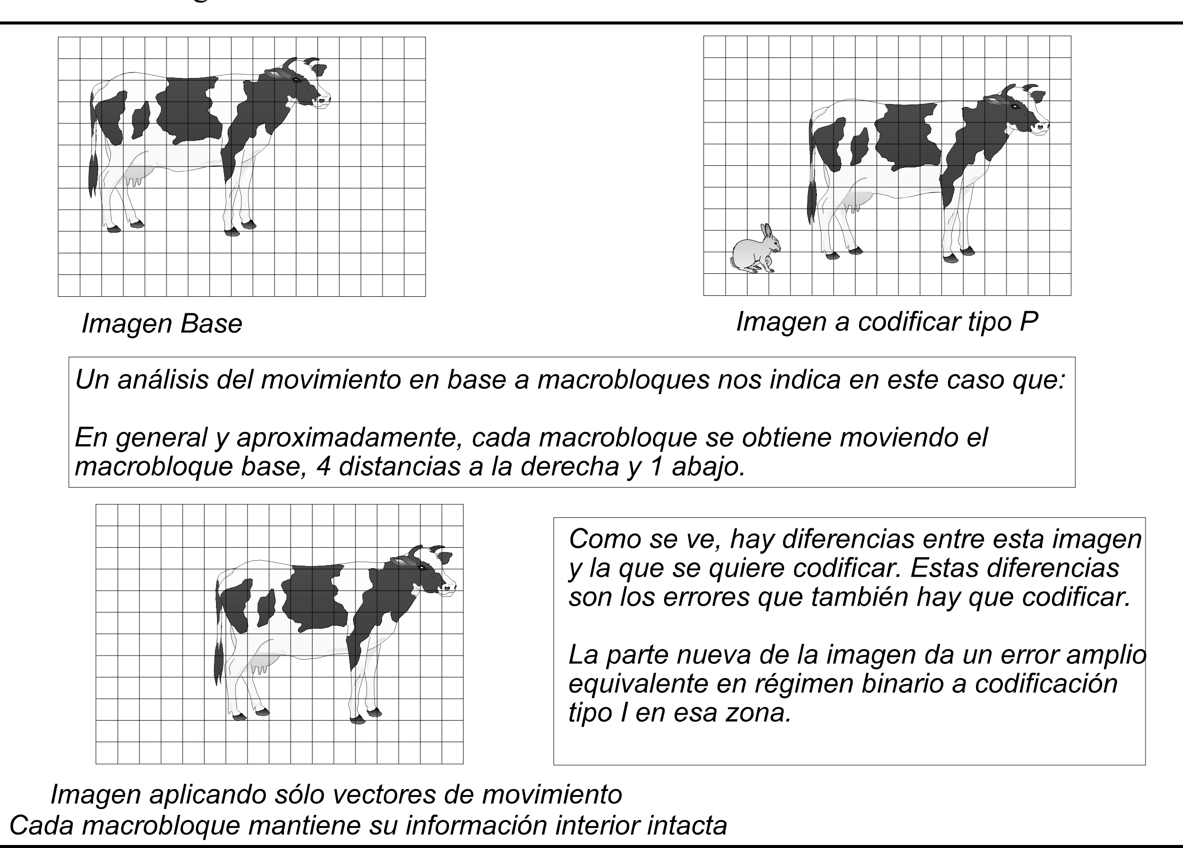


Fig.I-11.- Ejemplo de la limitación de la predicción tipo P.

Una solución para disminuir el error a codificar es poder realizar una compensación de movimiento desde una imagen posterior que ya contenga al conejo. Ese es el fundamento de la codificación predictiva Bidireccional.

Si la imagen a codificar va a ser tipo B, el proceso del codificador se complica, aunque al decodificador se le hacen las cosas más fáciles. La idea es que si una imagen se codifica en base a una imagen base anterior y a otra imagen base posterior, se pueden calcular los vectores de movimiento y la consiguiente "imagen error" tanto respecto a la imagen base anterior como a la imagen base posterior. Esto supone el doble de trabajo que para una codificación de imagen

tipo P. El interés radica en elegir para cada macrobloque de la imagen a codificar, qué imagen base lo puede obtener con menor error (y por tanto con menor información o número de bits). De esta manera, cada macrobloque genera en la codificación una información de vector de desplazamiento más la información de 16x16 píxeles de error (16x16 luminancias + 2x8x8 diferencias de color para 4:2:0) más la información de qué imagen base usar (la anterior o la posterior). Esta última información es muy corta y a cambio tenemos información de errores mucho menor que en el caso de codificación tipo P.

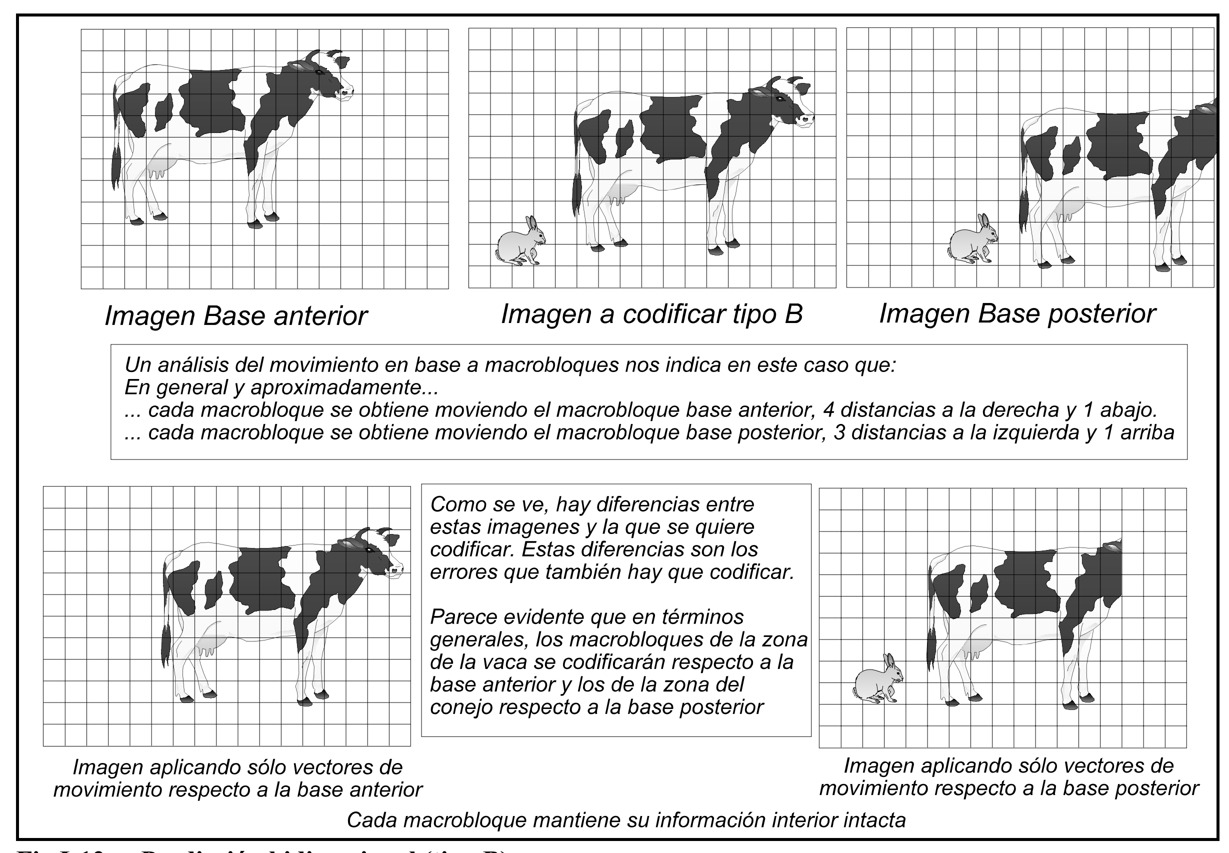


Fig.I-12.- Predicción bidireccional (tipo B)

El resto de detalles es igual que en codificación tipo P. El gran exito de este tipo de "imágenes" tipo B es principalmente en secuencias en que la imagen varía introduciendo nuevos elementos visuales y haciendo desaparecer otros. En ese caso, la codificación tipo P genera grandes errores en dichas zonas de la imagen, pues las imágenes base anteriores no tienen dicha información nueva. Sin embargo, las imágenes base posteriores usadas en tipo B, sí tienen dicha información nueva, por lo que generarán menor error en esas zonas.

Se parte de la evaluación de los errores de predicción y de los vectores de movimiento calculados con los valores de la imagen. Los errores de predicción se calculan en base a bloques y se transforman con DCT para seguir el mismo proceso de codificación explicado anteriormente, solo que lo que se codifica son valores de errores de predicción y no valores de la imagen. Además de los datos de errores de predicción, se envían los vectores de movimiento, calculados en base a macrobloques. Estos vectores no sufren transformación DCT. Ambas informaciones permiten la recuperación de imágenes a partir de otras, mediante "compensación de movimiento".

Como los vectores de movimiento son información esencial en estos procesos, conviene reducir su aportación al régimen binario, realizando una codificación diferencial o predictiva de los sucesivos vectores de movimiento en vez de mandar los datos absolutos de todos ellos. El dato que se codifica es el error de la predicción. Para no acumular errores, se reiniciará el predictor cada cierto tiempo. Los valores del error de predicción se codifican con VLC.

Selección de Slices, Imágenes, GOP's y Secuencias

Hasta aquí se ha descrito el proceso de codificación en las estructuras más básicas en las que tiene lugar. Como también se ha dicho, los datos obtenidos de esas estructuras se agrupan en estructuras mayores, que no sólo son estructuras de organización de datos sino que también tienen relación física con la imagen codificada.

Slices

Son agrupaciones de macrobloques ordenados en el sentido normal de barrido de la señal vídeo. Así, un slice de 10 macrobloques se relaciona con una parte de la imagen con una altura de 16 líneas y un total de 160 píxeles de ancho. Por ejemplo, si es un slice de una imagen tipo I, primero se tendrán los datos del primer macrobloque (6 bloques de 8x8 datos de luminancia y crominancia 4:2:0) correspondientes a 16 líneas de alto y 16 píxeles de ancho de la imagen. Estos datos se someten al proceso DCT+VLC y van precedidos de una cabecera de macrobloque.

Después se hace lo mismo con el siguiente macrobloque.

En ningún caso la estructura de slices afecta o se usa en la detección de movimiento sino sólo en la organización de datos en la codificación de la imagen original si se trata de una codificación de cuadro I o de la imagen error si se trata de una codificación de cuadro P ó B.

Los slices no tienen que ser todos iguales y de hecho no lo son. Su interés suele basarse en la limitación del efecto de los errores de transmisión, reiniciando los predictores de los valores que se codifican predictivamente. Esta reinicialización es conocida por el codificador y el decodificador y equivale a mandar un valor absoluto del dato para comenzar una secuencia de predicciones sin error acumulado.

Algunas reglas de selección de slices son las siguientes: Todos los macrobloques de un slice deben estar en la misma fila horizontal de la imagen (comparten las mismas 16 líneas de imagen). Los slices no se solaparán. Los slices pueden redefinirse de imagen a imagen. Los slices se codifican en el mismo orden en que van apareciendo en la imagen según el orden de barrido normal.

Imagen de cuadro o de campo

Además de seleccionar las imágenes en cuanto a tipo I, P o B, cabe este otro tipo de selección.

Se puede aplicar el proceso de estructuración de datos y procesos de compresión vistos a los datos de un cuadro de la imagen (p.e.720x576) o a los datos de un solo campo (720x288). En el segundo caso se obtendrán el mismo número de de imágenes, pero cada una de ellas con una ordenación y reordenación de datos diferente, al tener primero los datos de un campo y luego los del otro, teniendo que realizar el decodificador la imbricación de campos en un único cuadro.

Esta posibilidad es importante para obtener mejores resultados en algunos tipos de imágenes originales. Por ejemplo, cuando hay movimiento, es más fiable y fácil detectar el

movimiento analizando un solo campo que el cuadro completo, y luego aplicar la compensación de movimiento a ambos campos del mismo cuadro. También pueden conseguirse mejores resultados en resolución vertical de color al usar 4:2:0 si se aplica a campos y no a cuadros.

GOP o "Group of Pictures"

Un GOP es simplemente un conjunto de imágenes que se han codificado de forma que sólo la primera de ellas es tipo I, y las demás son predictivas tanto P como B.

De cara a una gran compresión, es interesante que el GOP sea largo, es decir, que haya mucha cantidad de imágenes predictivas frente a imágenes codificadas intracuadro. Para otras aplicaciones esto puede ser un inconveniente, pues debe recordarse que en el GOP todas las imágenes predictivas terminan necesitando la tipo I para su decodificación.

Ejemplo: (recuerde que las tipo P sólo se referían a imágenes base tipos P ó I)

Sea un GOP con 5 imágenes (Im1....Im5) que se codifican de la siguiente manera:

Im1... Tipo I (cuadro de datos I1)... Referida a sí misma
Im2... Tipo B (cuadro de datos B1)... Referida a Im1 e Im3
Im3... Tipo P (cuadro de datos P1)... Referida a Im1
Im4... Tipo B (cuadro de datos B2)... Referida a Im3 e Im5
Im5... Tipo P (cuadro de datos P2)... Referida a Im3

En el decodificador:
Se recupera la Im1 a partir de los datos de I1.
Los datos de P1 junto con Im1 permiten recuperar Im3.
Los datos de B1 junto con Im1 e Im3 permiten recuperar Im2.
Los datos de P2 junto con Im3 permiten recuperar Im5.
Los datos de B2 junto con Im3 e Im5 permiten recuperar Im4.

Tres conclusiones importantes:

1.-Sin Im1 obtenida a partir de los datos codificados I1 no se pueden recuperar las demás

imágenes Im2...Im5.

2.-Es distinto los "datos codificados de la imagen (I,Pn,Bn)" que la "imagen Imn", aunque sea normal denominar a los "datos codificados" con el nombre de "imagen". Esto ya debería estar claro.

3.-El orden con que se realiza el proceso de codificación y el de decodificación no coincide con el orden original de las imágenes.

Esta tercera es muy importante. Se puede ver que para codificar o decodificar la imagen 2 es necesario disponer en el codificador o haber decodificado ya en el decodificador, la imagen 3. Así, siguiendo el ejemplo, la codificación de estas 5 imágenes en un GOP del tipo "IBPBP" requiere una reordenación de las imágenes así: "Im1;Im3;Im2;Im5;Im4".

Esta reordenación nos impone un retardo mínimo de un cuadro en el proceso de codificación. Además, el codificador realmente trabaja con un GOP del tipo "IPBPB"

Parece lógico, y así lo refleja la propuesta del estándar MPEG-2, que dicho orden se mantenga en la transmisión de los datos del GOP hasta el decodificador. Se habla por tanto de una ordenación de tipos a realizar (presentar) (IBPBP) y una ordenación a transmitir (IPBPB).

El decodificador recibe las imágenes en ese mismo orden y las procesa recuperando las imágenes originales en un orden erróneo para la presentación ("Im1;Im3;Im2;Im5;Im4"). Lo que deberá hacer el decodificador es reordenarlas en el orden correcto. Para ello es necesario incluir en el GOP la información de ayuda correspondiente además de disponer de una memoria de varios cuadros en el decodificador.

Cada GOP puede tener distinta organización de tipos de cuadros.

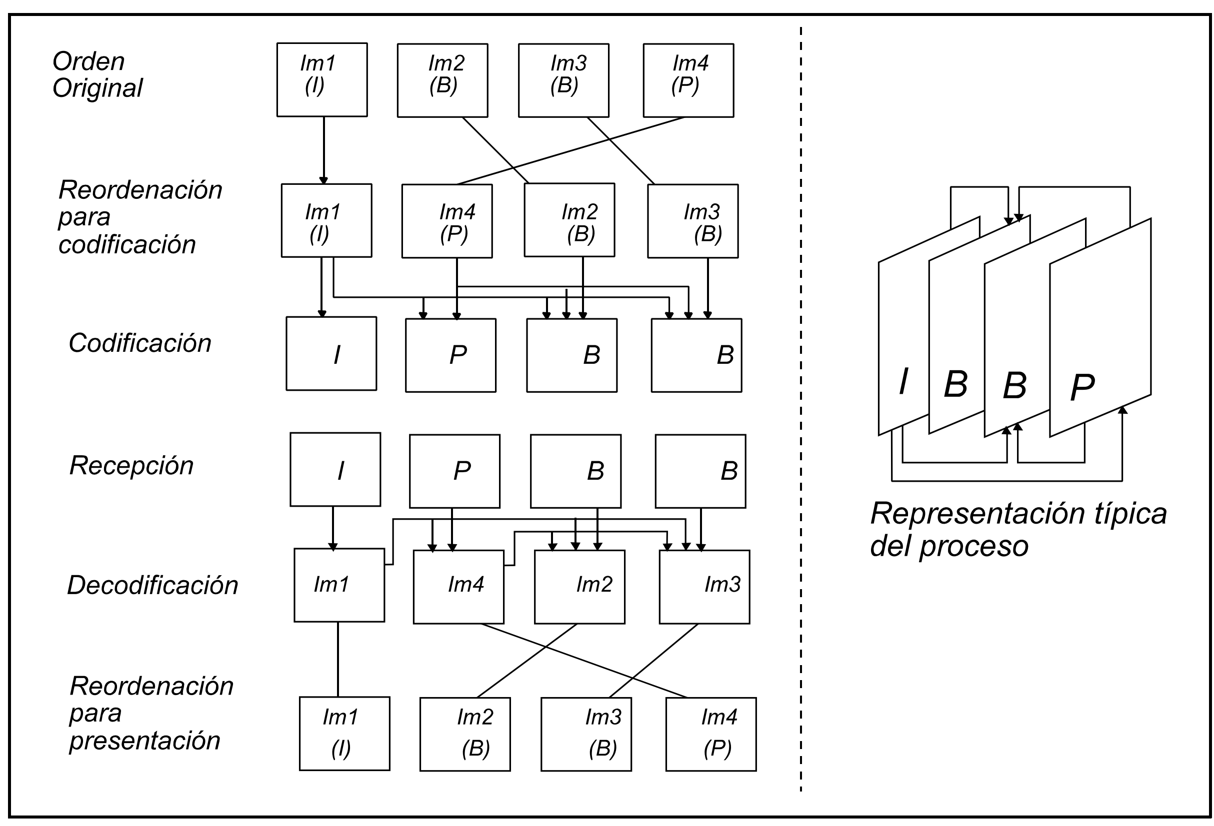


Fig.I-13.- Reordenación de las imágenes en la codificación

En realidad, la sucesión IPBB sólo aparece en el primer GOP de la trasnmisión, que además resulta ser más corto. El resto de GOPs reordenados vuelven a presentar sucesión IBBP, pero con los BB de posiciones originales muy anteriores.

Im	1	2	3	4	5	6	7	8	9	10	11	12	13
Codif	I1	B2	B3	P4	B5	B6	I7	B8	B9	P10	B11	B12	I13
Tx	I1	P4	B2	B3	I7	B5	B6	P10	B8	B9	I13	B11	B12
	GOP 1				GOP 2								

Existe la opción de 'GOP cerrado' en el que se impide que haya imágenes B anteriores a la imagen I que inicia el GOP.

Im	1	2	3	4	5	6	7	8	9	10	11	12	13
Codif	I1	B2	B3	P4	B5	P6	I7	B8	B9	P10	B11	P12	I13
Tx	I1	P4	B2	B3	P6	B5	I7	P10	B8	B9	P12	B11	I13
	GOP 1						GOP 2						

Para que el GOP sea cerrado, es necesario que la imagen anterior a una I sea P en vez de B.

Secuencia

Secuencia es el conjunto de GOP's que determinan una secuencia completa de vídeo. No tienen que coincidir con la duración completa del programa, pues la cabecera de secuencia incluye datos que pueden cambiarse a lo largo de un programa.

Además de estas indicaciones sobre los que son slice, imagen, GOP y secuencia, hay que tener en cuenta que cada comienzo de estas estructuras conlleva la existencia de cabeceras e información esencial para la adecuada reconstrucción por parte del decodificador de la imagen inicial.

CODIFICACIÓN MPEG-2 de AUDIO

La codificación MPEG-2 de audio no está dentro del enfoque de este texto, pero la existencia de esta señal no es insignificante. Concretamente, al tratar de las capas del sistema y de los flujos de programa y transporte, se considerarán existentes los 'Elementary Stream' de la capa de compresión de audio.

Básicamente se trata de una compresión basada en técnicas perceptuales que hacen uso de modelos psicoacústicos para realizar la asignación de bits. El análisis se realiza mediante subbandas. Los aspectos concretos de la compresión están referidos a 3 capas (en vez de llamarse perfiles), que son Layer I, II y III, que ya existen en MPEG-1. Un decodificador MPEG-2 es capaz de decodificar señal comprimida MPEG-1 de la misma capa.

Mientras que MPEG-1 sólo comprime señal estéreo, MPEG-2 permite la compresión de sonido multicanal 3/2 más uno opcional de mejora en baja frecuencia (LFE) con señal de hasta 120 Hz, configurando un sistema tipo 5.1. La señal MPEG-2 debe ser compatible hacia atrás, de modo que un decodificador MPEG-1 sea capaz de extraer una señal estéreo completa del programa. Esto supone organizar la señal comprimida en un aparte compatible y una extensión para recuperar el sonido multicanal. Para ello, en codificación se debe realizar un matrizado de canales que debe deshacerse en el decodificador. Estas operaciones afectan a la separación entre canales resultante. MPEG-2 también contempla el uso de un sistema no compatible hacia atrás ('NBC') denominado AAC ('Advanced Audio Coding') que no permite la recuperación de la señal estéreo en un decodificador MPEG-1.

Existen otras estrategias de codificación de audio multicanal que no son conforme a MPEG-2 aunque pueden compartir el mismo tren de datos de 'transport stream' MPEG-2, como son Dolby Digital (Dolby AC-3), DTS o SDDS. En estos casos, como en el AAC, el decodificador recupera los canales por separado, sin señal estéreo compatible. En el caso de no tener un sistema de reproducción multicanal, el decodificador debe incorporar un circuito final de matrizado a estéreo o a Dolby Prologic, que siga las recomendaciones de ITU-R.BS.775.

FLUJO DE DATOS ELEMENTAL (E.S.) Y POR PAQUETES (P.E.S)

"Elementary Stream (E.S.)"

Las descripciones previas han establecido la manera de codificar la información de vídeo consiguiendo un régimen binario menor. También se ha presentado la existencia de un método con la misma finalidad para audio. El resultado es una secuencia de bits para vídeo y otra secuencia independiente para audio. Ambas siguen unas normas concretas. Esta estructura de tren de datos serie ininterrumpido se denomina genéricamente "Elementary Stream E.S." y existe uno para cada información de vídeo o audio de un "programa audiovisual". Un E.S. lleva datos correspondientes a la denominada "capa de compresión" de MPEG-2.

Sin embargo, no está pensado que la conexión entre equipos de la señal MPEG-2 se haga simplemente con este tipo de tren de datos. Téngase en cuenta, que incluso en el caso más simple de un sólo programa con una imagen y un sonido asociado, se tiene una E.S. de vídeo y otra E.S. de audio independientes, que se multiplexan temporalmente para su transmisión o uso; y debe asegurarse que la posterior reproducción sea síncrona, respetando la simultaneidad original entre la imagen y el sonido.

Si la situación es tal que se tiene a la vez varios programas con sus correspondientes E.S., los detalles de asociar cada E.S. a su programa original y de sincronizar los vídeos y audios, se complican requiriéndose la definición de una estructura de organización de los E.S. para su transmisión o uso. Esta nueva estructura es la denominada **"capa de sistema"** de la señal MPEG-2.

Es importante definir dos conceptos usados por la capa del sistema para referirse a ciertos contenidos de la capa de compresión: se llama "Unidad de presentación" a la información correspondiente a una imagen original completa o bien a la información de un "frame" AES/EBU y se llama "Unidad de acceso" a la información codificada, según MPEG-2, de cualquiera de

dichas unidades de presentación. Estos conceptos son útiles para diferenciar aspectos del proceso de decodificación y de la sincronización.

Debido a la versatilidad de MPEG-2, la capa de sistema no sigue una estructura única sino que puede seleccionarse entre ciertas posibilidades. En cualquiera de los casos, la estructuración de señal hace uso de "paquetes de señal". Así, todos los E.S. pasan por un proceso de "paquetización" dando lugar a los denominados P.E.S. ("Packetized Elementary Stream").

"Packetized Elementary Stream"

Los PES se obtienen directamente de cada ES, de modo que existen tantos PES como ES. Los diversos PES son en principio independientes unos de otros. Cada PES sólo contienen información de un E.S. concreto. La información de datos de un PES consiste en los bytes de la E.S. tomados de forma contigua. Sólo la aparición de las cabeceras de los paquetes hace que exista diferencia entre el contenido de un PES y de su ES.

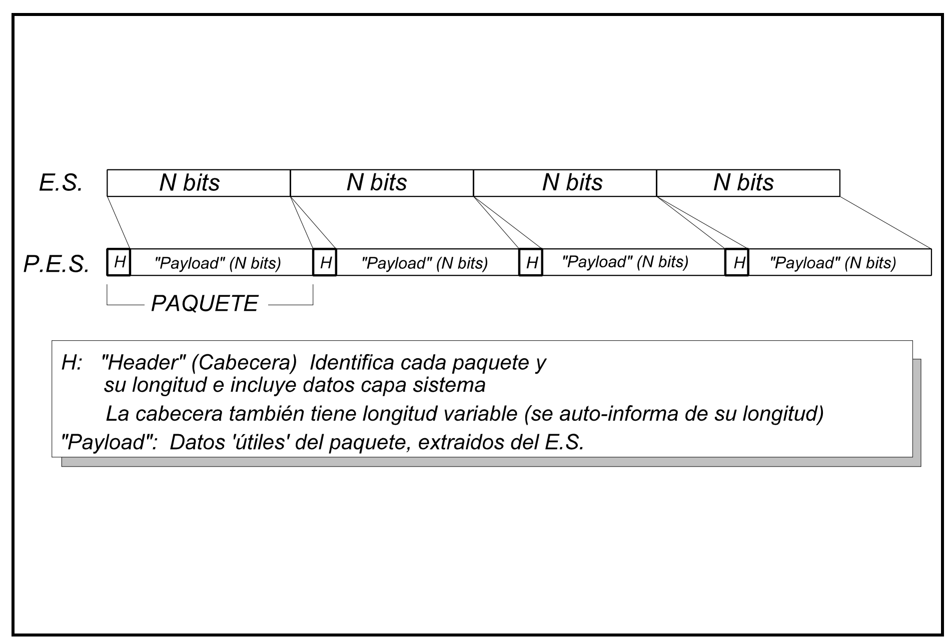


Fig.I-14.- Estructura y obtención de los PES

No es necesario que la parte de datos ("payload") de un paquete PES comience con un código de comienzo de alguna estructura de la capa de compresión (secuencia, GOP, slices,...).

Toda esta nueva información que se incluye en las cabeceras ya no pertenece a la capa de compresión sino a la de sistema. Desde el punto de vista del sistema, se puede decir que los PES son una subcapa de la "capa de sistema", aunque a su vez contengan la capa de compresión. Para completar la capa de sistema, y por tanto la transmisión de la señal MPEG-2, deben combinarse todos los PES en un único tren de datos. Esta combinación puede hacerse de dos maneras, dando lugar al denominado "Transport Stream (TS)" o bien al denominado "Program Stream(PS)".

Existen casos en que sólo interesan los PES de un sólo ES (p.e. sólo vídeo). En este último caso se pueden obviar tanto el TS como el PS y usar un simple "PES Stream" para conectar equipos.

CAPA DE SISTEMA

La importancia de esta capa de estructuración de las señales digitales está en permitir combinar uno o más E.S. de vídeo y audio, del mismo o de distintos programas, además de otros posibles datos, en una única señal digital serie adecuada para ser transmitida o almacenada. Téngase en cuenta que cada ES tiene, en general, una frecuencia de muestreo y un régimen binario independiente.

Multiplexación y sincronización

Existen dos conceptos esencialmente básicos en la organización de diversos ES: multiplexación y sincronización. La información de los distintos E.S. es básicamente multiplexada en una señal o estructura mayor. Los problemas pueden surgir en el receptor, una vez demultiplexadas las informaciones correspondientes a cada programa, al tenerlas que presentar al usuario con la debida sincronización entre ellas. Para asegurar esta función con éxito, la estructuración de la señal en el codificador debe seguir estrictas normas e incluir informaciones de identificación y temporización.

Estas informaciones aparecen de forma aparentemente redundante (no lo es realmente) en las dos subcapas de la capa de sistema, existiendo tanto en los PES como en los "transport o program streams" formados con dichos PES.

Subcapa PES

Para la sincronización, los PES incorporan campos de datos de **ESCR** ("Elementary Stream Clock Reference") que marcan referencias de la señal digital del PES, y que con el campo **ES_rate** ("Elementary Stream Rate") permite calcular el tiempo asociado a cada byte del PES. También incorporan los campos **PTS** ("Presentation Time Stamp") y **DTS** ("Decoding Time

Stamp") referidos a los momentos de las unidades de presentación y de acceso completas.

El **ESCR** indica momentos concretos que debe cumplir la señal digital PES a su llegada al decodificador y permite comprobar la sincronización de la señal que llega con respecto al reloj patrón del sistema reproducido en el decodificador. Concretamente, el ESCR consiste en el dato codificado del tiempo en que se codificó el último byte de dicho ESCR. Esta codificación se hace en función del número de ciclos del reloj patrón del codificador y no en unidades temporales como el "segundo".

El **DTS** marca pautas al decodificador al respecto del momento de realizar la decodificación de una parte de la señal ("unidad de acceso") respecto a otras. Por ejemplo, cuando se recibe un cuadro tipo P, debe esperarse a recibir un mínimo de datos (concretamente la "unidad de acceso") antes de empezar a decodificar. El dato DTS también es un valor temporal concreto, codificado como el número de ciclos de reloj patrón en ese momento.

El **PTS** marca pautas relativas a la presentación de las imágenes decodificadas ("unidades de presentación"). Por ejemplo, el orden real de las imágenes no tiene que coincidir con el orden en que se han codificado y decodificado. Es evidente que existen ciertas relaciones entre DTS y PTS para cada tipo de imagen codificada. La forma de dar esta información temporal es igual que las anteriores.

En todo momento, la información temporal que se incluye en el tren de datos se hace en función de números de ciclos de reloj patrón (t/T_r) y no unidades de tiempo.

Por supuesto, todos estos datos temporales son obtenidos a partir de un reloj patrón del sistema existente en el codificador y usados respecto al reloj patrón del sistema existente en el decodificador. Además, no todos los paquetes deben contener todos estos datos de referencia temporal, siendo bastante que los contengan un numero suficiente de paquetes por segundo.

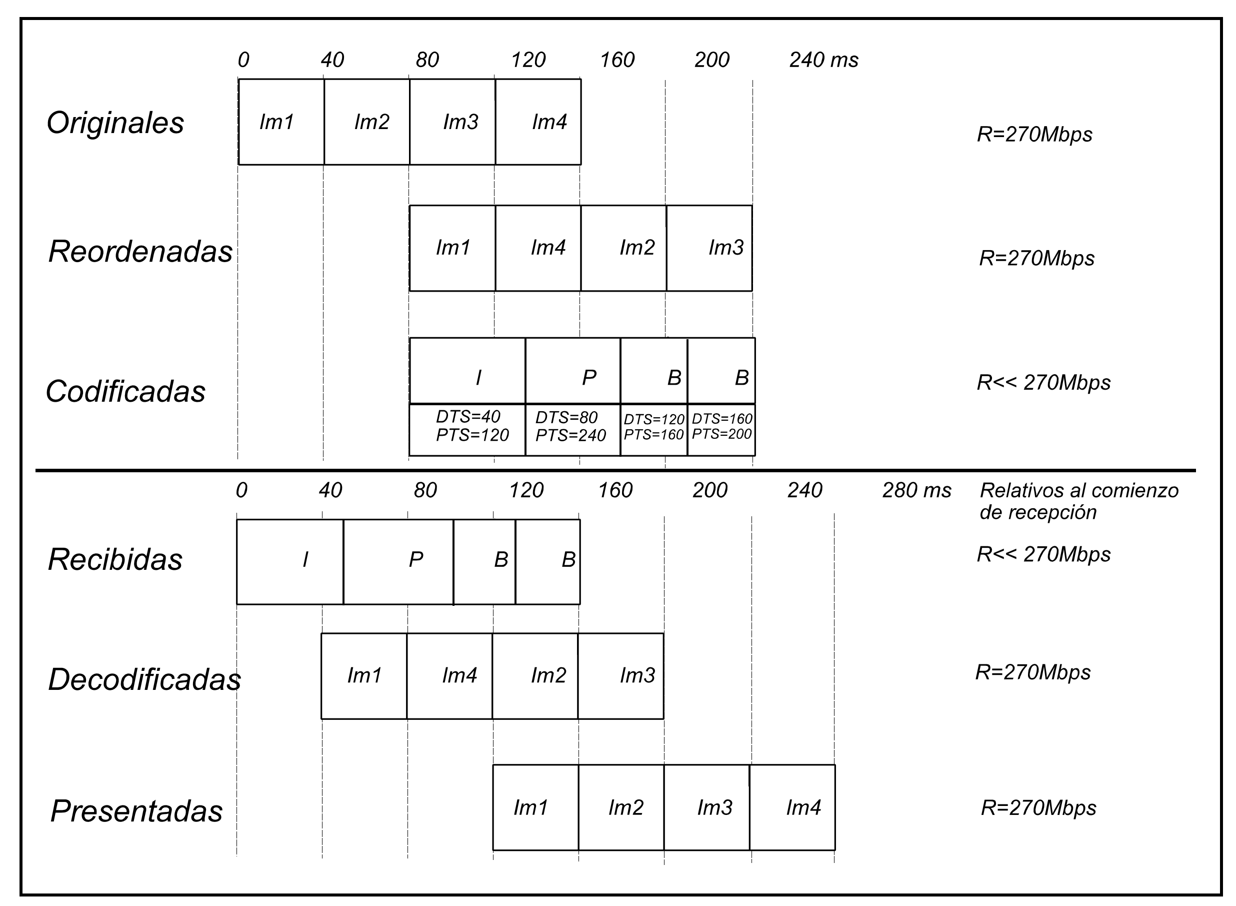


Fig.I-15.- Ejemplo del uso de las marcas temporales

Subcapa de paquetes (PS o TS)

Los paquetes PES rara vez se usan tal cual para transmitir señal de programas completos. Como ya se ha indicado, estos paquetes se organizan dentro de una estructura mayor que puede llamarse "Program Stream" o "Transport Stream" según el método usado. Las normas y datos para esta agrupación constituyen la "subcapa de paquetes" de la capa de sistema. Se llama así porque tanto el PS como el TS son señales basadas en paquetes específicos, con los que se envían adecuadamente los PES.

En los siguientes apartados se profundiza en los detalles de estas señales que constituyen la señal digital MPEG-2 utilizable para aplicaciones.

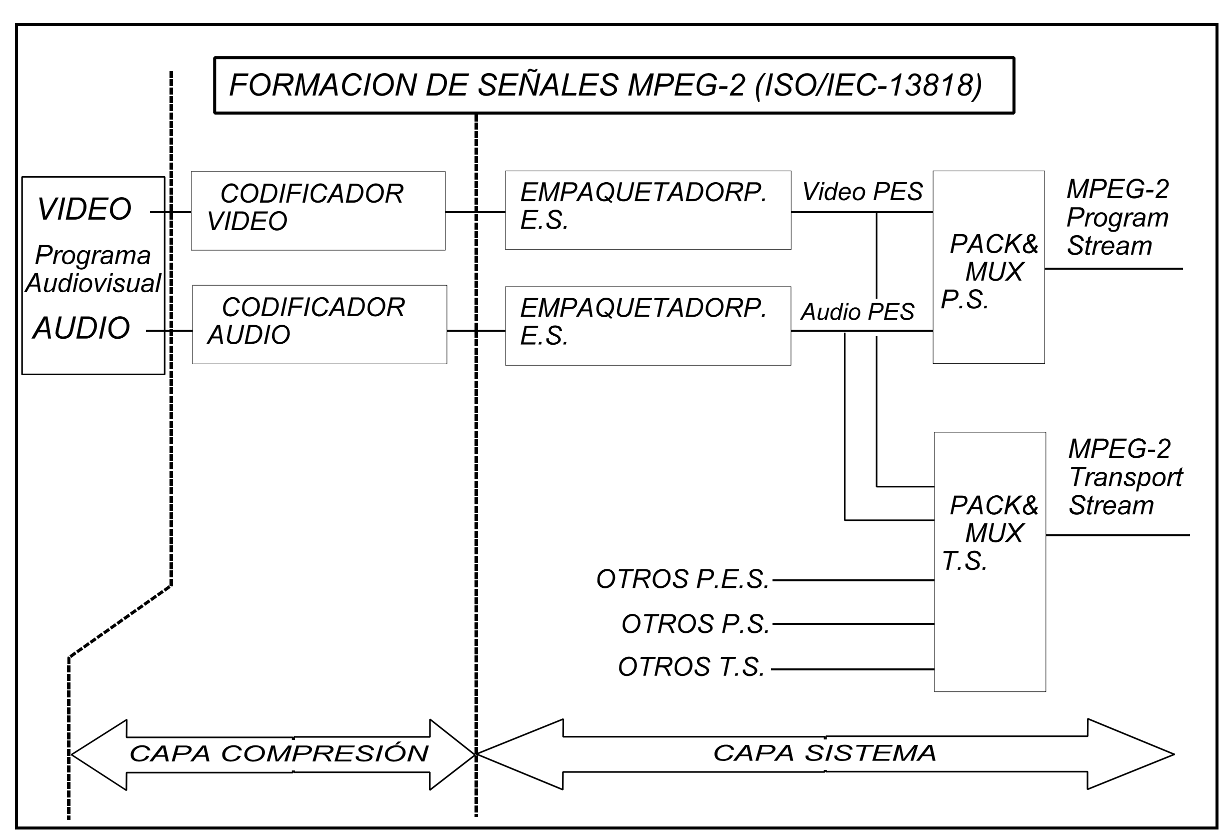


Fig.I-16.- Esquema general de la generación de señales MPEG-2

FLUJO DE DATOS DE PROGRAMA ("Program Stream")

Introducción

El PS ("Program Stream") es la estructura de datos usada para enviar o almacenar información de 1 programa (vídeo+audio+datos) codificado en MPEG-2, en entornos y dispositivos en los que es improbable que aparezcan errores en la señal. El PS puede ser de régimen binario fijo o variable según lo sean los PES que lo forman.

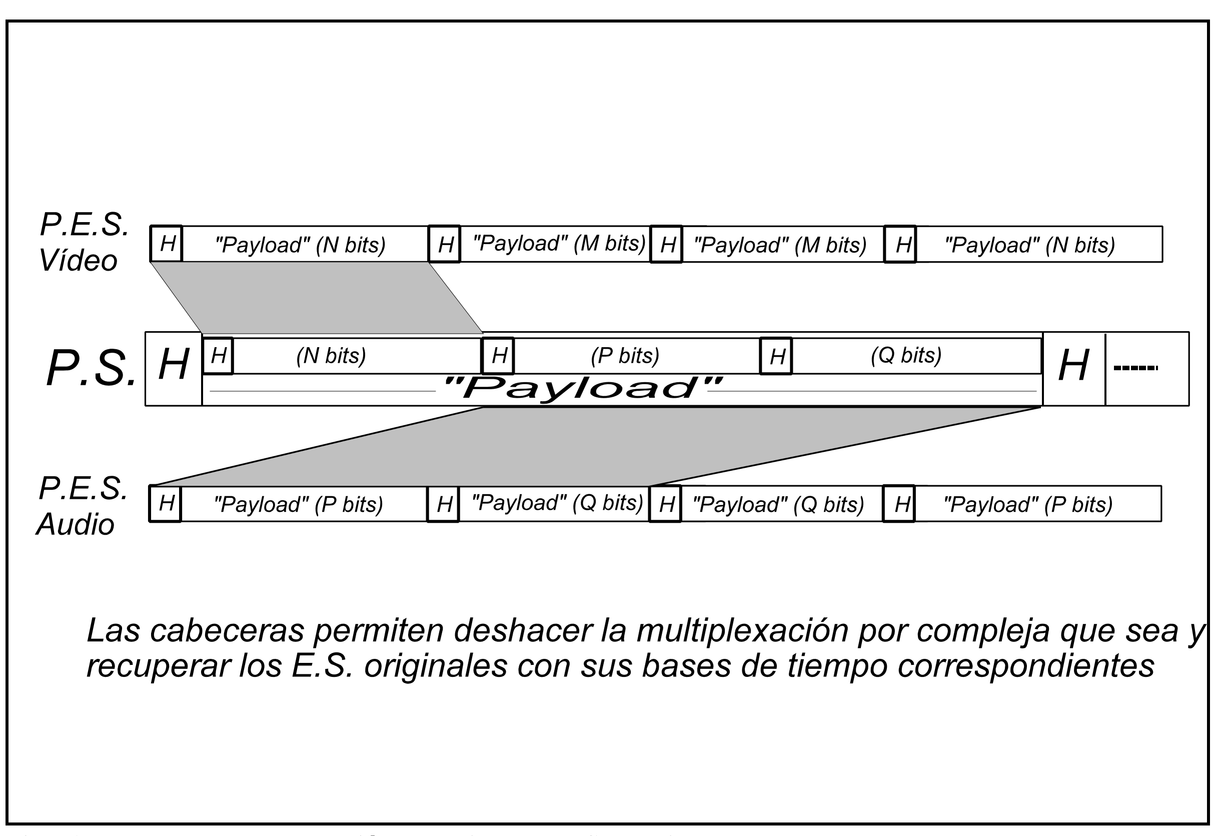


Fig.I-17.- Estructura y obtención de un 'Program Stream'

El PS multiplexa los datos de todos los P.E.S. (vídeo, audio, datos) correspondientes a un único programa. El PS se organiza en nuevos paquetes ("paquetes PS"), cuya longitud no tiene que ser fija, consistentes en una cabecera y un "payload" de uno o varios PES (también pueden ser cero PES). La cabecera de estos paquetes PS incluye información de tiempos (**SCR: "System Clock Reference")** y del régimen binario en el paquete (**PMR: "Program Mux**

Rate").

En la cabecera existe además la posibilidad de incluir datos y parámetros del sistema aplicables al PS.

El concepto general de un decodificador de PS es el de la figura.

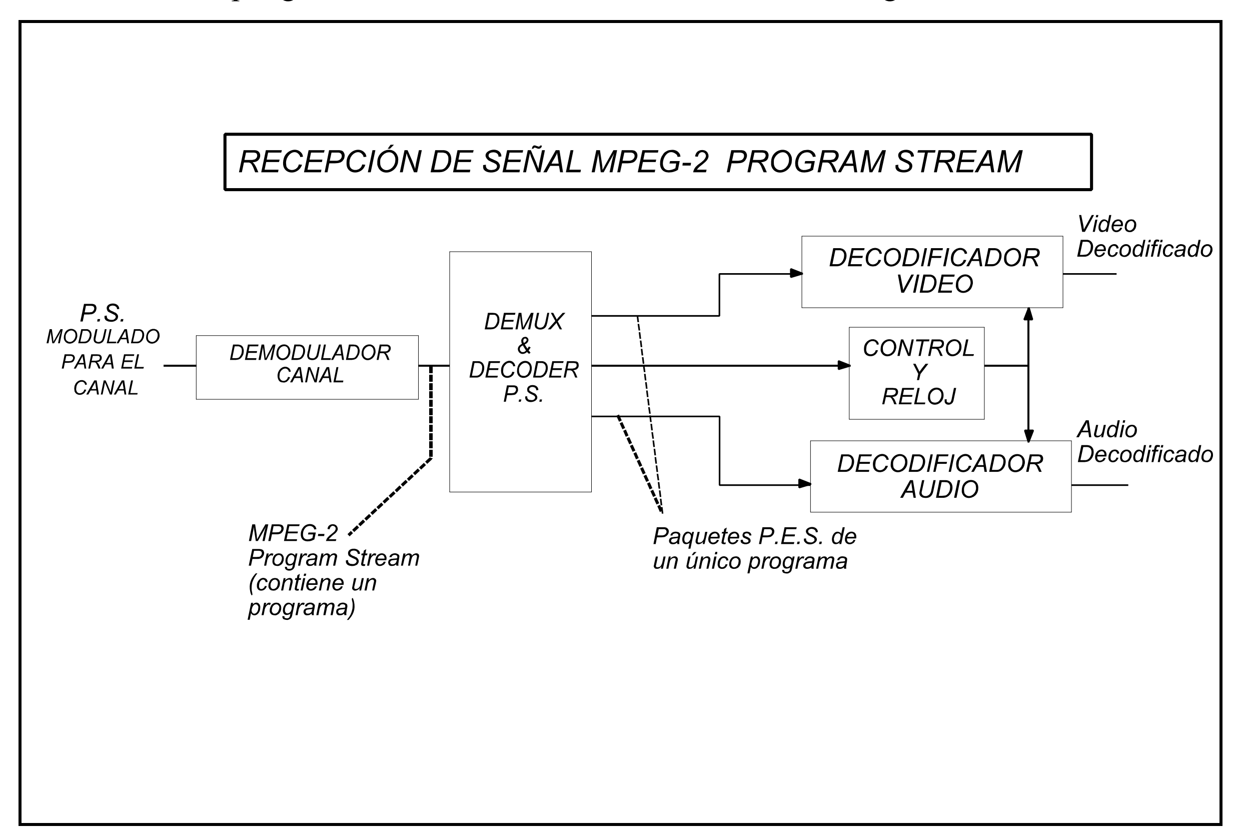


Fig.I-18.- Esquema general de decodificador de PS

Decodificación y referencias temporales

La recomendación ISO/IEC define un decodificador hipotético conocido como el decodificador modelo para el sistema (P-STD "Program Stream-System Target Decoder"). El uso de este decodificador permite definir ciertos puntos clave en la decodificación, que deben ser tenidos en cuenta en la codificación, introduciendo información adecuada en la capa del sistema. En absoluto es una indicación sobre el diseño de decodificadores reales.

Cualquier suceso que ocurra en el codificador en un momento de tiempo, debe ocurrir en el decodificador en el mismo momento, salvando la existencia de un retardo que será el mismo para todos los sucesos a considerar.

La referencia del momento en que suceden ciertos sucesos en el codificador se incluye como datos en la capa del sistema (unos en los PES y otros en el PS).

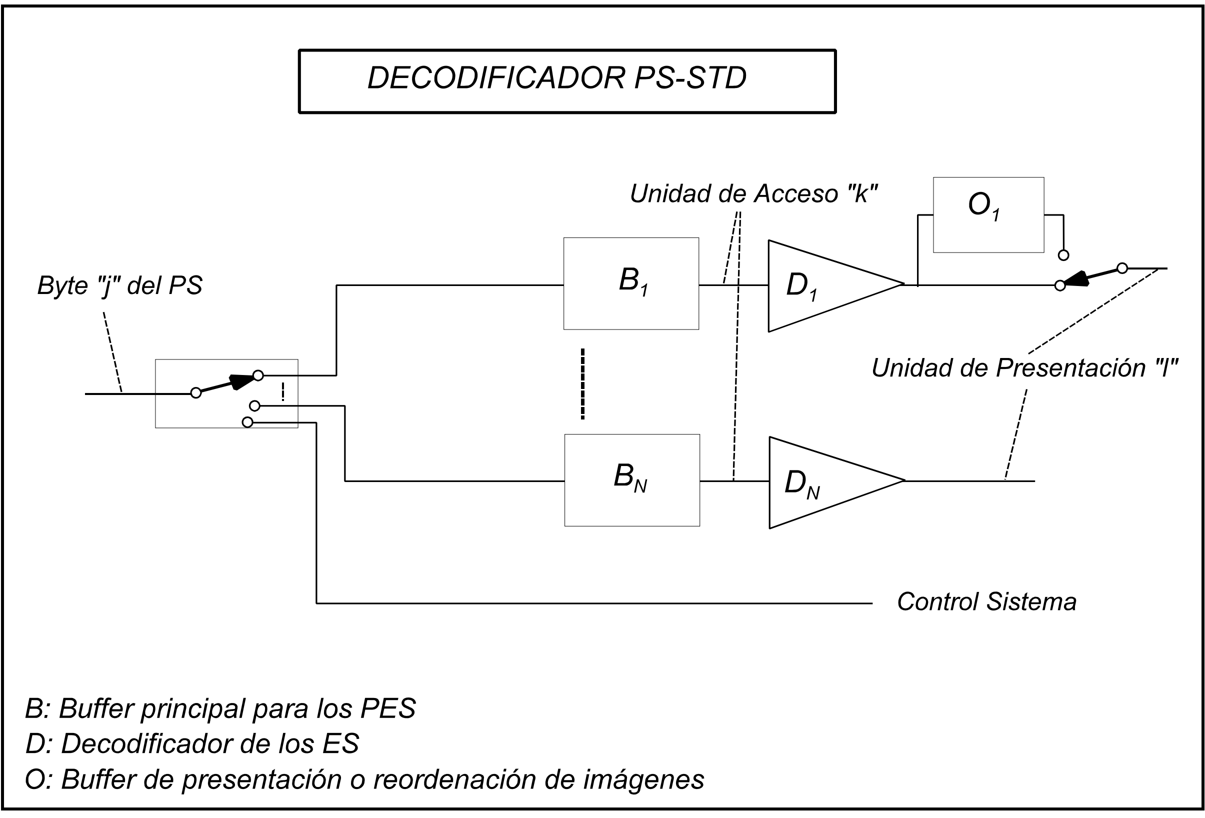


Fig.I-19.- Modelo conceptual de referencia de decodificador de PS

El reloj del sistema que controla la base de tiempos de todo el proceso es de 27MHz con una variación máxima de ±540Hz con una velocidad máxima de variación de 0.075Hz/sg. El dato de este valor se denomina SCF ("System_Clock_Frequency").

Los datos que se codifican en el sistema permiten calcular instantes de tiempos concretos que deben respetarse en el decodificador.

El SCR existente en la cabecera del paquete PS está codificado como dos datos (SCR_base (33 bits) y SCR_extension (9 bits)) que se combinan para dar el SCR con la ecuación:

SCR(i)=300*SCR_base(i) + SCR_ext(i)

cuyo valor indica el número de ciclos de reloj patrón correspondientes a dicho momento "i".

El último byte que lleva información del SCR se considera el byte de referencia temporal dentro del paquete PS y es el que se denomina con el índice "i" , y el tiempo que le corresponde se calcula como:

t(i)=SCR(i)/SCF (SCF: System_Clock_Frequency)

El tiempo correspondiente a cualquier byte (denominado con índice "j") debe calcularse a partir del de referencia "i" mediante la fórmula:

$$t(j) = \frac{SCR(i)}{SCF} + \frac{j - i}{50 * PMR}$$

siendo SCF el dato "System_Clock_Frequency", y PMR el "program_mux_rate". La diferencia "j-i" es el número de bytes que separan al byte "j" del byte de referencia temporal "i" del paquete PS. Como se ve, t(j) depende no sólo de la referencia del reloj sino de otros parámetros del sistema que deben incluirse en el PS al codificarlo.

Para que estas fórmulas funcionen, el codificador debe codificar los datos de SCR a partir del dato de t(i) en la codificación de la siguiente manera:

SCR_base(i)=((SCF*t(i)) DIV 300)%2^{33} codificable con 33 bits

SCR_ext(i) = ((SCF*t(i)) DIV 1)%300 codificable con 9 bits

Nótese que la operación DIV es realmente "división tomando el valor entero del resultado" y que el símbolo % indica la operación "Módulo".

Este aspecto de la identificación temporal es esencial para las operaciones de sincronización durante la presentación de los distintos ES recuperados (imágenes y sonidos).

Decodificación y memoria buffer

Como ya se vió al estudiar la codificación, los datos de la capa de compresión necesitan de un buffer tanto a la salida del codificador como a la entrada del decodificador para permitir flujos elementales de régimen binario constante a partir de un proceso de compresión de datos con régimen binario variable. Esta necesidad es aplicable a cada E.S. Por otro lado, el uso de PES incluye la necesidad de incluir una cabecera en el codificador y la de quitarla en el decodificador. Además, un PS tiene requerimientos adicionales al combinar un PES de vídeo y un PES de audio que ocupan tiempos distintos en la señal digital, pero que son "simultáneos" en la presentación (sincronización audio-vídeo). Todos estos procesos requieren también el uso de memoria buffer en la que introducir los datos que llegan y sacar los datos de interés hacia cada parte del decodificador.

Volviendo al diagrama del P-STD, la función de separación del tipo de datos, la realiza un demultiplexor genérico. Las salidas de dicho demultiplexor son básicamente tres:

-"Unidades de acceso" de vídeo.

-"Unidades de acceso" de audio.

- Datos de temporización y control del sistema.

Los datos de temporización y control del sistema incluyen todos los recibidos en las cabeceras de sistema, ya sean de paquetes PS o de paquetes PES. Cada uno de ellos será usado en una parte adecuada del proceso de decodificación.

Las "unidades de acceso" pasan a distintos buffer, libres de cabeceras, es decir, sólo incluyen datos útiles de la señal codificada (capa de compresión), pero llegan al buffer en intervalos definidos por la llegada de paquetes. Se considera en el P-STD que el acceso de cualquier byte "j" desde la entrada hasta el buffer que le corresponde, es instantáneo y por tanto entra en el buffer en el instante t(j). Se llama "tiempo de decodificación (td)" al momento en que todos los datos del buffer se usan para decodificar y obtener la "unidad de presentación" correspondiente. Cuando llega ese momento, el buffer debe haber recibido todos los datos que

componen una unidad de acceso completa del E.S. El P-STD considera que esta decodificación es instantánea.

En el caso de las "unidades de presentación" de vídeo, su orden no coincide con el orden en que llegan sus datos en las unidades de acceso, por lo que se requiere una reordenación de imágenes que se realiza con la ayuda de un buffer de ordenación.

Evidentemente, este funcionamiento del P-STD sólo indica la filosofía. Los decodificadores reales no son instantáneos y usarán las soluciones circuitales que consideren oportunas, para conseguir la reproducción adecuada de todas las informaciones sin problemas de sincronización y sin problemas de saturación o vaciado de los buffer.

Las necesidades de buffer en la decodificación son previsibles en la etapa del codificador, por lo que los datos sobre los tamaños de buffer convenientes son mandados en el PS.

PES de datos especiales en PS: "Program Stream Map"

Ciertos datos especiales no contenidos en las cabeceras de paquetes PES ni en la de los paquetes PS, son enviados conjuntamente como si se tratase de un PES más. En la cabecera de dicho tipo de PES se activa, con el valor adecuado, un campo de identificación ("Stream_id").

Estos datos se clasifican en dos grupos:

-Program Stream Map

-Program Stream Directory

El **PS-Map** incluye descripciones de las ES que lo componen y de cómo se relacionan entre ellas. Para ello incluye los denominados **DESCRIPTORES**.

El **PS-Directory** incluye un directorio de las "unidades de acceso", en el mismo orden en que aparecen en el PS, indicando el tipo de dato que lleva, su longitud, los PES que llevan su información y el PTS ("Presentation Time Stamp") correspondiente. También indica si se trata de una imagen tipo I y de donde buscar información sobre los parámetros de codificación/decodificación. Si la unidad de acceso lleva datos de audio, el directorio localizará las palabras de sincronismo de los "frames AES/EBU"

Descriptores

Los descriptores que aparecen en los PES de "PS-Map", pueden referirse ya sea al "program stream(PS)" o a algunos de los "elementary stream (ES)" que lo componen una vez hechos paquetes PES. A continuación se presentan los descriptores.

***Video stream descriptor**

Es un conjunto de bits que proporcionan información sobre la codificación de los datos del ES de vídeo. Entre esos datos se encuentran:

-Frecuencia de cuadros del vídeo

-Bandera indicando si los datos son codificados según MPEG-2 o según MPEG-1

-Perfil y nivel de la compresión MPEG-2

-Formato de codificación de la crominancia usado (4:2:0,4:2:2,...)

***Audio stream descriptor**

Este conjunto de bits aportan datos sobre la codificación del ES de audio, especialmente el referido al "layer" usado (I, II o III).

***Hierarchy descriptor**

Permite identificar las ES que llevan diversos componentes de una codificación jerárquica de vídeo y/o audio. La codificación jerárquica se refiere normalmente a las codificaciones con escalabilidad o a la organización especial de datos conocida como "particiones". Si se usa una codificación de ese tipo, cada capa jerárquica constituye un ES

distinto y se multiplexan en el PS. Recuerde el lector la referencia sobre "capa base" y "capa de mejora" hecha en el apartado de "niveles y perfiles".

Se indica el tipo de jerarquía (o escalabilidad), listado o índice de las capas jerárquicas existentes y qué ES componen las diversas capas jerárquicas de un mismo tipo de información. También se indican qué capas jerárquicas son prioritarias (y cuales se pueden llegar a perder sin afectar definitivamente a la señal).

***Registration descriptor**

Permite identificar formatos especiales de codificación de datos para uso privado. El lector debe entender que el Program Stream es una estructura de transmisión digital que permite suministrar datos además de la información audiovisual, en el mismo PS. Estos datos no tienen porqué estar directamente relacionados con el contenido audiovisual.

***Data stream alignment descriptor**

Indica el tipo de alineamiento usado en el ES descrito. Cuando algún PES correspondiente al ES presenta en su cabecera una cierta bandera activada, indica que los datos de su "payload" pertenecen a una estructura definida por este descriptor. Los valores previstos son bastante genéricos: "picture","GOP", "Sequence" y "Audio Frame".

***Target Background grid descriptor y Video Window descriptor**

La combinación de estos dos permite obtener una presentación de la imagen en un área inferior a la total del monitor. El primero de ellos define la retícula o "grid" que se usará para ajustar la imagen en el monitor. El segundo indica la posición de la parte superior izquierda de la imagen dentro de la retícula definida antes. También define la prioridad de la ventana, por si se estuvieran recibiendo varias.

***Conditional access descriptor**

Evidentemente es información para controlar el acceso condicional a la información audiovisual, que tendrá algún tipo de "scrambling". La documentación ISO/IEC 13818 no entra en aspectos concretos del acceso condicional.

***ISO 639 lenguage descriptor**

En muchos programas audiovisuales suele asociarse un ES de datos auxiliares para ayudas a discapacitados auditivos o visuales. Estos datos siguen las normas ISO 639 e ISO 8859 y suelen ser textos o sonidos de ayuda en uno o varios idiomas. Este descriptor indica los idiomas así como el tipo de ayuda contenida en los PES del ES descrito.

***System Clock descriptor**

Este descriptor informa sobre el reloj patrón usado para generar las referencias temporales en los PES y en el PS. Indica si se ha usado un reloj patrón exterior (que quizá esté también disponible para el decodificador) y la precisión del reloj. Si la precisión del reloj es mejor que las 20ppm requeridas (27MHz±540Hz), se puede codificar su valor de precisión en este descriptor.

***Multiplex buffer utilization descriptor**

Cuando se activa, indica la máxima variación pico-pico del retardo de multiplexación de todos los paquetes del PS. También indica la estrategia a seguir para el uso de los buffers.

***Copyright descriptor**

Su uso permite identificar propietario de los derechos del programa. También puede aplicarse a cada ES.

***Maximun bitrate descriptor**

Indica el régimen binario máximo del PS. Sólo indicar que se codifica en múltiplos de 50 bytes/sg. También se puede optar por describir el dato para el PS completo o para cada uno de los ES.

***Private data indicator descriptor**

Permite describir lo que sea sobre datos privados. No está definido por ISO/IEC, salvo que su longitud es de 32 bits de datos precedidos de 16 bits de valores prefijados.

FLUJO DE DATOS DE TRANSPORTE ("Transport Stream")

Introducción

De forma análoga al "Program Stream", el "Transport Stream (TS)" agrupa en paquetes TS los paquetes PES. El TS además permite la agrupación de datos provenientes de varios programas distintos (incluso teniendo cada uno de ellos regímenes binarios distintos) y está pensado para entornos y sistemas en los que existe posibilidad de errores en la señal digital. Dichos errores pueden manifestarse como errores de bits o como pérdidas de paquetes completos.

El TS puede tener régimen binario fijo o variable, y en cualquiera de los dos casos, los ES que lo constituyen pueden ser a su vez de régimen binario fijo o variable. Sin embargo, en la práctica es casi imposible tener un TS de régimen variable cuando contiene varios programas.

Su máxima potencia se ve clara cuando se tiene en cuenta que un TS puede formarse no sólo a partir de señales elementales (ES) sino también puede formarse a partir de "Program Streams" o de otros "Transport Stream" ya existentes, que a su vez contienen varios programas.

El diseño del TS permite realizar con el mínimo esfuerzo una serie de operaciones de interés, entre las que se encuentran:
*Recuperar los datos codificados de un sólo programa de todos los que lleva el TS, para su decodificación y presentación de las señales originales.
*Extraer los paquetes TS de un sólo programa de los que lleva, para formar un nuevo TS que sólo lleve ese programa.
*Extraer del TS los datos de un sólo programa y formar un "Program Stream" con dicho programa.
*Convertir un PS en un TS para pasar por un entorno con errores y recuperar al final el mismo PS inicial u otro PS válido con el contenido del programa inicial.

De forma análoga al PS, el TS tiene dos capas: la de sistema y la de compresión. La decodificación de un TS se realiza según una concepción análoga a la de un PS, como aparece en la figura:

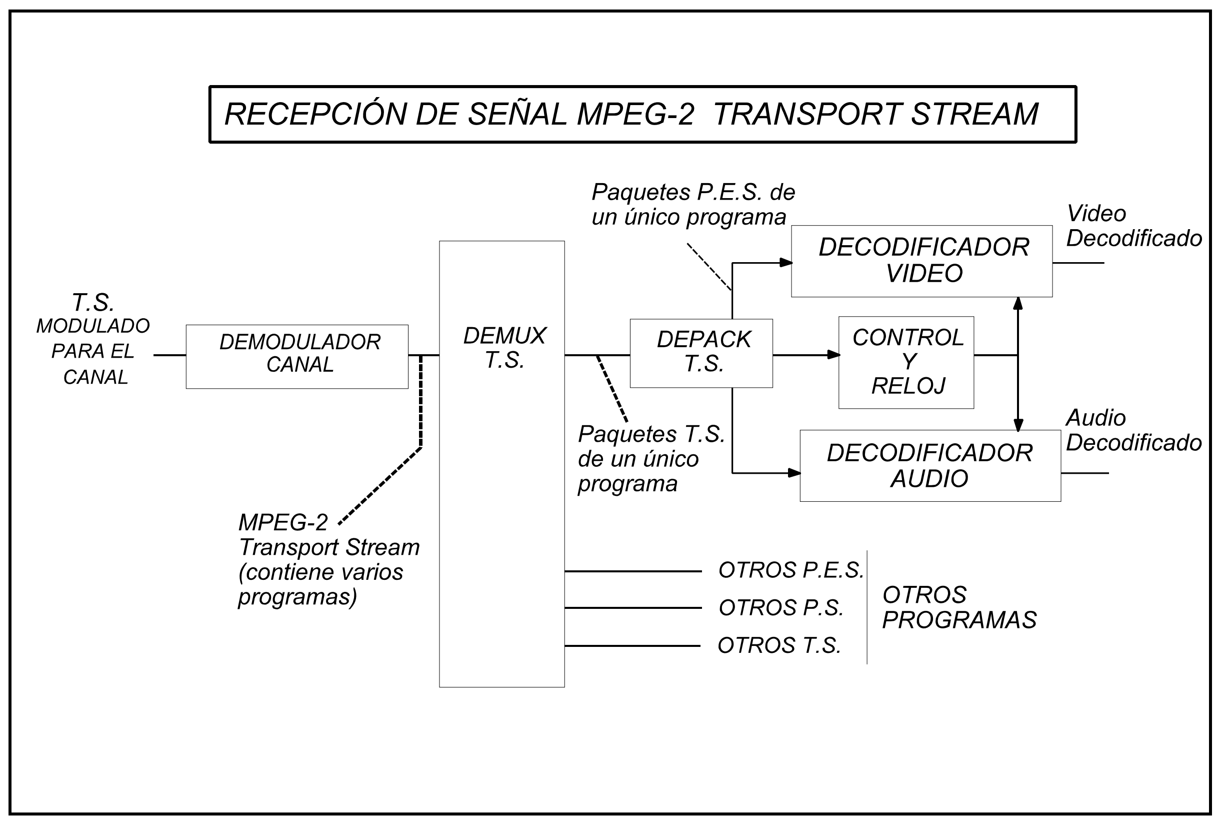


Fig.I-20.- Diagrama general de decodificador de 'Transport Stream'.

Una vez más, este diagrama es conceptual y no tiene que coincidir con decodificadores reales.

La capa de sistema del TS también se divide a su vez en dos subcapas: la de paquetes TS (para operaciones de la señal completa y la multiplexación de programas y ES) y la de paquetes PES (para operaciones referidas a cada ES específica).

La sincronización de datos exige que en cada programa exista una referencia de tiempos propia, que se llamará **PCR (Program Clock Reference)** cuyo uso es equivalente al SCR existente en el "Program Stream".

Contenido de un "Transport Stream"

El TS es una única señal digital que contiene uno o más programas. Cada programa está constituido a su vez por una serie de señales elementales (ES) que comparten una misma base de tiempos y que pueden ser de vídeo, audio o datos. Esas ES aparecen dentro de paquetes (PES) que contienen información para la presentación sincronizada de dichas ES. Además de los programas, el TS incluye multiplexados otros datos.

Cada ES contiene sus "unidades de acceso", que son la representación codificada de las "unidades de presentación". La "unidad de presentación" para vídeo es la imagen (cuadro) y para audio es un "audio frame AES/EBU". Las "unidades de acceso" incluyen todos los datos codificados de una imagen o de un "Frame AES/EBU", y además incluyen los códigos de inicio de la estructura MPEG que inicia dicha imagen. Así, si la imagen es inicio de secuencia, la "unidad de acceso" incluye la cabecera de secuencia. Si la imagen es inicio de GOP, incluye la cabecera de GOP. Si es una "simple" imagen, contendrá el código de inicio de imagen. Para audio, la unidad de acceso incluye la palabra de sincronismo del "frame AES/EBU". Todo esto corresponde a la capa de compresión y ya se ha visto.

Los ES se agrupan en paquetes PES que incluyen su cabecera propia y a los ES como "payload". La cabecera PES identifica el ES e incluye referencias temporales para la decodificación (DTS ("Decoding Time Stamp")) y para la presentación (PTS ("Presentation Time Stamp")) de las unidades de acceso contenidas.

Los PES se insertan en paquetes del "Transport Stream" que también tienen su cabecera propia y como "payload" incluyen los PES. La inclusión de PES en los paquetes TS se hace sin romper las cabeceras PES. Para asegurar esto, el primer bit de la cabecera de un PES siempre se colocará como el primer bit de un "payload" TS (pero no viceversa).

A pesar de esta estructura tan aparentemente anidada, cada paquete TS sólo lleva información de una sóla ES. No debe olvidarse que las estructuras basadas en paquetes no exigen la anidación de estructuras inferiores, y que eso depende de la longitud del paquete de nivel superior respecto a la longitud de los paquetes de nivel inferior. De hecho, pueden hacer

falta varios paquetes TS para recuperar la información de un único paquete PES, ya que como se verá los paquetes TS son de corta longitud. **Esta es una gran diferencia con el "Program Stream"**

Los paquetes TS empiezan con una cabecera en la que se incluye un identificador de paquete PID ("Paquet ID"). Dicho identificador debe usarse junto con la información contenida en las tablas de información específica de programa (PSI "Program Specific Information"). Las tablas PSI se llevan en el TS como si se trataran de otras PES (aunque no con la misma estructura). Existen 4 tablas que suministran la información necesaria y suficiente para la demultiplexación y presentación de los programas:

-Program Association Table (PAT)

-Program Map Table (PMT)

-Network Information Table (NIT)

-Conditional Access Table (CAT)

La segunda de ellas (PMT) es la que lleva la lista de PID's (identificadores de paquetes TS) correspondientes a cada programa. También incluye la lista de los paquetes TS que incluyen entre sus datos los PCR ("Program Clock Reference") de cada programa.

Los paquetes TS también pueden ser de relleno, es decir, pueden existir paquetes TS sin datos útiles, cuya utilidad es rellenar para cumplir las exigencias de régimen binario del canal o entorno. Estos paquetes pueden ser quitados cuando no sean necesarios y en ningún caso aportarán datos a los ES del decodificador.

Decodificación: referencias temporales y memoria buffer

Igual que para PS, existe un decodificador hipotético para el TS (T-STD), que aparece en la figura. Una vez más, recordar que es sólo un diagrama "filosófico".

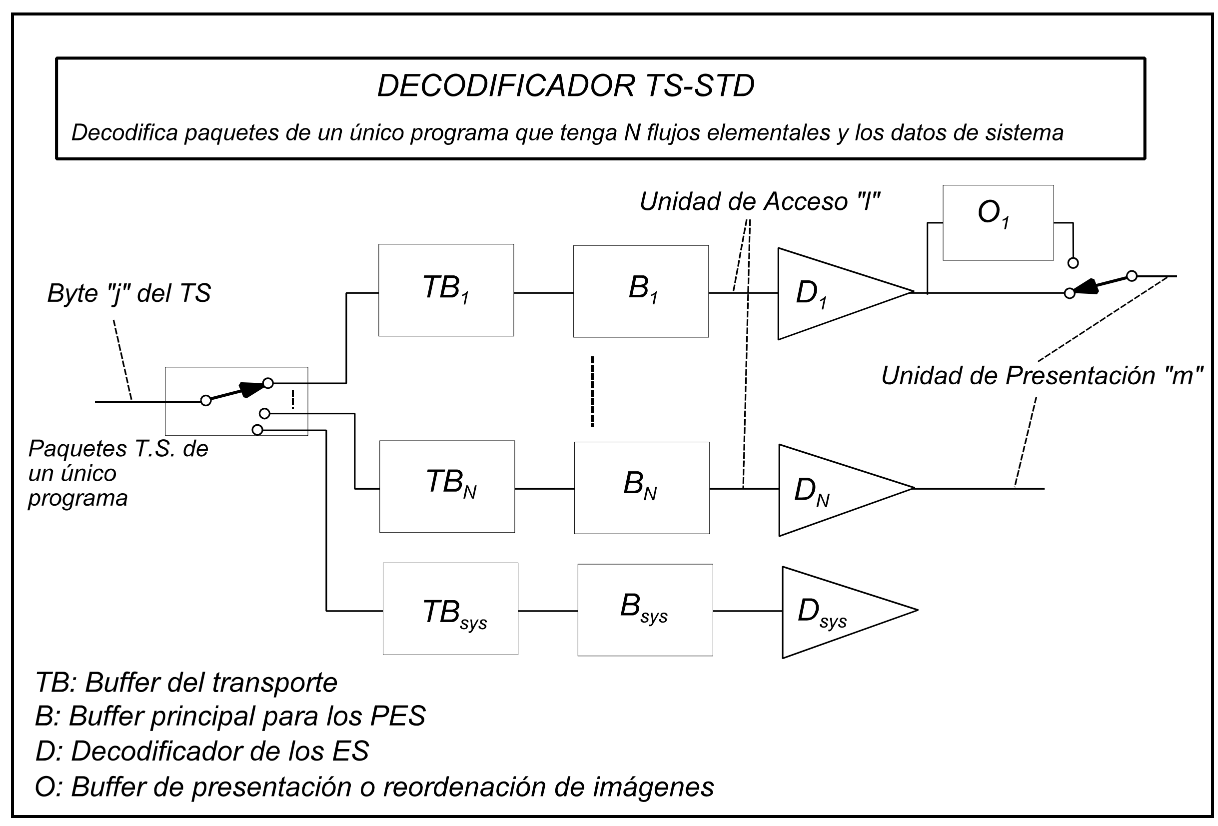


Fig.I-21.- Esquema conceptual de referencia de decodificador de TS

Para la frecuencia del reloj del sistema, es aplicable lo indicado en el PS: el reloj del sistema que controla la base de tiempos de todo el proceso es de 27MHz con una variación máxima de ±540Hz con una velocidad máxima de variación de 0.075Hz/sg. El dato de este valor se denomina SCF ("System_Clock_Frequency").La identificación temporal en el TS se realiza con los datos codificados del PCR en vez del SCR usado en PS, pero el planteamiento, uso y significado es el mismo. La única salvedad es que el TS al llevar varios programas, tiene unos PCR distintos para cada uno.

Sin embargo, el T-STD sólo decodifica un programa en cada momento, luego su base de tiempos es única en cada momento y hace uso de un único conjunto de PCR.

Con los PCR y las fórmulas siguientes, se identifica el momento de tiempo t(j) en que un byte entra en el decodificador.

$$t(j) = \frac{PCR(i)}{SCF} + \frac{j - i}{TR(j)}$$

PCR(i)=300*PCR_base(i) + PCR_ext(i)
SCF: System_Clock_Frequency
TR(j): es el régimen binario del transporte ("Transport Rate")

El dato de "Transport Rate" (Régimen binario del TS) se calcula del modo siguiente:
*Número de bytes en el TS que separan los bytes en los que está incluido el último bit del dato PCR de dos códigos PCR consecutivos y del mismo programa...
...dividido por...
*...diferencia entre los valores de tiempo codificados en dichos PCR.
que se corresponde con la fórmula:

$$TS(j) = \frac{(k - i) * SCF}{PCR(k) - PCR(i)}$$

siendo "i" el índice del PCR recibido anteriormente al byte "j" y siendo "k" el índice del primer PCR recibido posteriormente al byte "j". Los momentos de llegada de dichos PCR son: t(i)=PCR(i)/SCF y t(k)=PCR(k)/SCF.

Una vez más, el PCR existente en la cabecera del paquete TS está codificado como dos datos (PCR_base (33 bits) y PCR_extension (9 bits)) que se combinan para dar el PCR con la ecuación:
PCR_base(i)=((SCF*t(i)) DIV 300)%2^{33} codificable con 33 bits
PCR_ext(i) = ((SCF*t(i)) DIV 1)%300 codificable con 9 bits
Nótese que la operación DIV es realmente "división tomando el valor entero del resultado" y que el símbolo % indica la operación "Módulo".

El manejo del buffer según se describe en el T-STD es como sigue. El paquete TS que

lleva datos de un ES indicado por el PID (p.e. el "n"), se transfiere completo al buffer de transporte "n". El paso de cada byte "j" a este buffer es instantáneo, de modo que el momento en que el byte j llega al buffer de transporte n es t(j).

Los paquetes TS que entran en dicho buffer de transporte n pueden ser datos útiles del ES n, datos repetidos ("Duplicate TS paquets") o incluso datos de paquetes TS de relleno. En este buffer se separan los datos que son para control del sistema y se eliminan los de relleno y se seleccionan cuales de los duplicados se usan. Los datos útiles de programa se transfieren al buffer principal n.

Cuando llegan PES consistentes en datos y tablas para el sistema, se llevan a un buffer de transporte específico. El lector interesado en conocer los tamaños de buffer y las velocidades de transferencia entre ellos, puede consultar la documentación ISO/IEC.

En el momento de decodificación de cada ES ($td_n(l)$), el buffer principal de dicha ES del programa que se está decodificando estará lleno con la unidad de acceso completa. Este momento se obtiene a partir de la información DTS. La decodificación se realiza instantáneamente según este T-STD y el buffer se vacía. La unidad de presentación obtenida se visualiza en el momento $tp_n(m)$ establecido por los PTS, haciendo uso si es necesario de los buffer de ordenación. En el caso de vídeo se considera que la presentación de todo el cuadro de imagen es simultáneo y ocurre en $tp_n(m)$. Para audio, ese momento de tiempo se considera como el inicial y el resto de las muestras de audio se presentan secuencialmente.

Como puede verse, el T-STD es un modelo filosófico orientado no a la realidad de los procesos sino a la definición de la secuencia de trabajo durante la decodificación y a la necesidad de buffer y sincronización.

Paquetes TS

En este apartado se va a profundizar algo más en la estructura y datos de los paquetes de "Transport Stream". La longitud de un paquete TS está normalizada siempre a 188bytes, de los cuales 4 bytes (32bits) son de cabecera y los otros 184 bytes (1472bits) son para "payload".

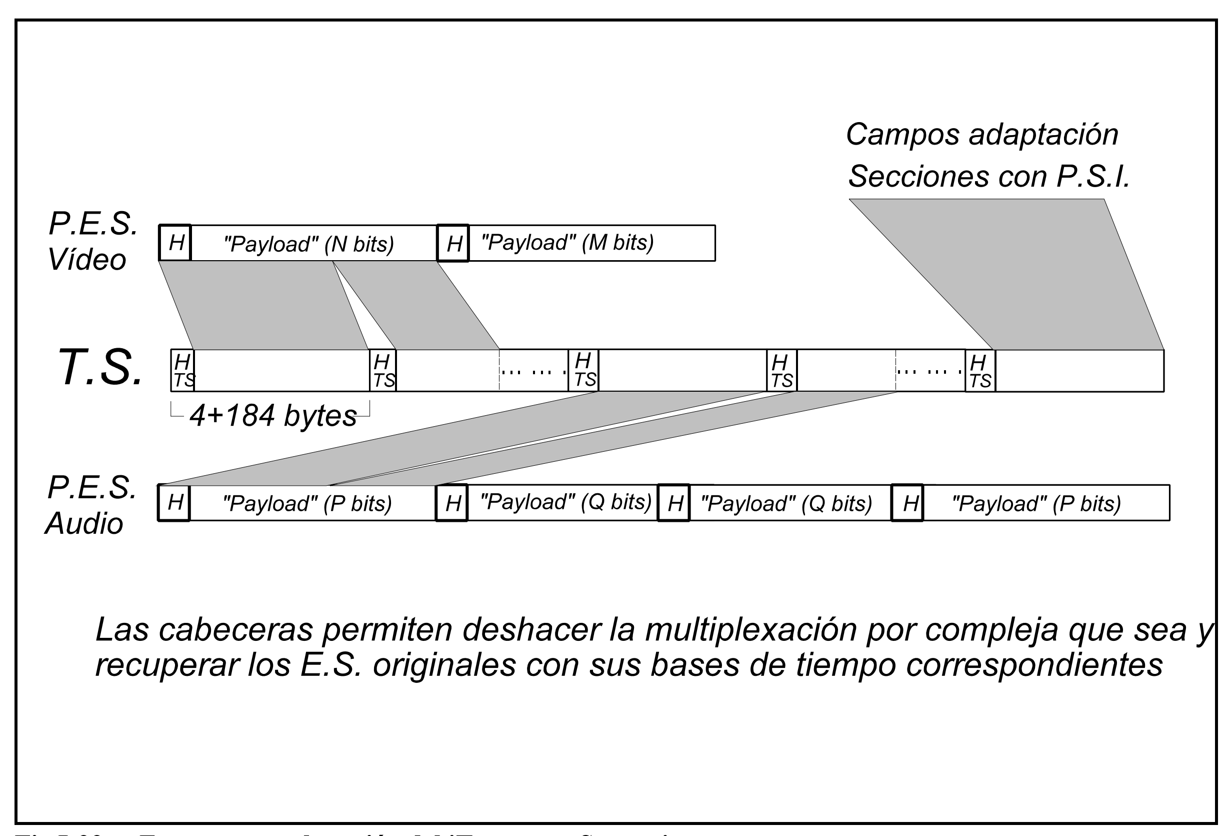


Fig.I-22.- Estructura y obtención del 'Transport Stream'

En la tabla de la página siguiente se indican los datos que se encuentran en la cabecera. Con esto se completan los 32 bits de la cabecera del paquete TS. A continuación se tiene el "payload". Este puede consistir en lo siguiente:

*Datos de un PES (correspondientes a un ES) o datos de PSI. Ambos casos se conocen como "contenido payload"

*Datos especiales sobre los PES, conocidos como "campos de adaptación".

Cabecera de un paquete TS		
*sync_byte	8 bits	Palabra fija para marcar el comienzo del paquete TS de forma inequívoca.
*Transport_error_ indicator	1 bit	Bandera que se activa si existe al menos un error no corregible en el paquete. Esta detección del error la harán equipos adecuados, para avisar a los siguientes equipos de la presencia del error.
*Payload_unit_ start_indicator	1 bit	Bandera que se activa si el payload empieza por la cabecera de un PES o bien si empieza por el "pointer_field" de un PSI, según que el contenido del paquete sea un PES o un PSI.
*Transport_priority	1 bit	Bandera que indica que el paquete es de máxima prioridad respecto a otros paquetes con el mismo PID (del mismo ES).
*PID	13 bits	Es un número entero entre 0 (H0) y 1594322 (H1FFF) que permite identificar el tipo de datos del paquete. Algunos de los valores están estandarizados para identificar algunas de las tablas de PSI indicadas anteriormente, para identificar paquetes de relleno y para usos futuros (valores reservados). El resto se pueden usar libremente para identificar ES.
*Transport_ scrambling_ control	2 bits	Indica la existencia de scrambling en los datos del payload, si son de la capa de compresión.
*Adaptation_field_ control	2 bits	Indica si el payload del paquete contiene datos de compresión y/o datos de "campo de adaptación". Sobre esta diferencia hablaremos más adelante.
*Continuity_counter	4 bits	Es un entero de 0 a 15 que se incrementa indicando la cuenta de paquetes TS con PES del mismo PID. Al llegar a 15 vuelve a empezar por 0.

Un paquete TS puede contener ambos tipos de datos o sólo uno de ellos. Esto queda indicado en la cabecera del paquete ("adaptation_field_control").

En el campo de adaptación encontramos los datos del PCR y otros que el lector puede consultar en la documentación ISO/IEC.

Si cualquiera de las dos informaciones no llena los 184bytes del payload del paquete TS, este se rellena con bytes sin valor ("stuffing bytes").

Campos de adaptación

Los campos de adaptación son actualmente 10, existiendo cierto espacio reservado para ampliaciones futuras. De los 10 campos, tres son sólo banderas y el resto son datos. Estos últimos también tienen asignadas otras tantas banderas.

Banderas sólo:

*Discontinuity_indicator	Se activa para indicar la discontinuidad en los valores de PCR respecto a los anteriores recibidos del mismo PID. Normalmente se debe al uso de una nueva base de tiempos a partir de este paquete.
*Random_access_indicator	Su activación indica que el siguiente PES del mismo ES, contiene un dato PTS y un punto de acceso al ES. Se llama punto de acceso al ES al primer byte de la cabecera de secuencia en vídeo o al primer byte de un "frame" en audio.
*elementary_stream_priority_ indicator	Indica prioridad del contenido ES de este paquete TS respecto al de otros paquetes TS.

Datos con banderas son:

*PCR	"Program clock reference".
*OPCR	"Original Program Clock Reference". Usado para reconstruir programas que han pasado por varios "Transport Streams". Sólo existe este dato si también existe el dato PCR.
*Splice_countdown	Si en los paquetes TS con mismo PID va a existir un cambio del programa original (punto de corte o empalme), se activará esta cuentaatrás que irá apareciendo en cada uno de los paquetes indicando cuantos paquetes TS faltan hasta el punto de corte. El último paquete, el que tenga la cuenta a cero, deberá contener el final de la "unidad de acceso" en curso.
*Transport private data	Usado para datos privados.
*Legal time window *Seamless splice. *Piecewise rate	Estos tres últimos aparecen agrupados como "extensión del campo de adaptación". Los dos primeros tienen que ver con manejo y accesos a los buffers y el tercero es una medida del régimen binario entre dos paquetes TS del mismo PID que tengan este campo activado.

PSI ("Program Specific Information")

Esta información incluye datos de la normativa y datos privados que permiten la demultiplexación de los programas por parte de los decodificadores. Cada programa está compuesto de uno o varios ES cada uno de los cuales está identificado con un PID. Estos programas o algunos de sus ES pueden tener acceso condicionado y estar sometidos a scrambling. Sin embargo, los PSI nunca estarán sometidos a scrambling.

En un "Transport Stream", los PSI se clasifican en 4 tablas de datos que se envían en varios paquetes TS, formando "secciones".

Cada una de estas secciones de datos incluye al final un código de redundancia cíclica para comprobar que no existe error en los valores recibidos.

Además de PSI, es posible enviar datos privados dentro de esta categoría, haciendo uso de ciertas secciones.

Las secciones son de longitud no fija y su comienzo queda indicado por un "puntero" en el "payload" del paquete TS que lleva PSI. La pertenencia de una sección a una cierta tabla se identifica mediante el dato "table_id" existente en cada sección.

Cada sección indica su número de orden y el número total de secciones que completan la tabla en cuestión. De esta manera es fácil detectar si se ha perdido alguna sección.

Las tablas pueden irse actualizando según cambian los programas, por lo que todas las secciones llevan un indicador de "versión" que permite eliminar las informaciones viejas. Este indicador es realmente un simple contador y no pretende identificar nada más.

***Program Association Table (PID=H00)**

Todo TS debe llevar al menos un paquete con este PID, conteniendo por tanto esta información, que da un listado de todos los programas que lleva el TS, indicando el valor de PID asignado a cada tabla "Program Map Table" que definen cada uno de los programas del TS. Por tanto, cada programa de la lista lleva asociado un PID en el que se busca la "Program Map Table", y que se fija en el codificador, sin existir valores específicos para ellos.

Además, se incluye el PID de los paquetes que llevan la "Network Information Table", que tampoco tiene un PID específico.

***Program Map Table**

Esta tabla contiene la información de la relación entre los programas y los ES que los componen. De esta manera, todos los programas del TS quedan definidos en esta tabla. La tabla completa se transmite dividida en secciones.

Es importante indicar que ninguna definición completa de un programa ocupa más de una sección. Así, cada sección lleva el indicador del número de programa al que se refiere.

Se indican en esta tabla, para cada programa:
*el PID de los paquetes TS que llevan los PCR.
*los ES que lo componen, indicando tipo de datos de cada ES y el PID de los paquetes TS que llevan cada uno de los ES.
*se incluyen **descriptores** de programa y de ES, que son los mismos que se indicaron en el apartado del "Program Stream".

***Conditional Access Table**

Esta tabla sí tiene un PID específico y debe existir siempre que exista algún programa con "scrambling", para indicar el método para recuperar la información original. También se divide en una o más secciones.

Cada sección incluye un conjunto de **descriptores de acceso condicional** como información específica. Este descriptor se mencionó de manera simple en el apartado de "Program Stream". Sólo basta ampliar aquí que incluye para cada programa con acceso condicional, el PID de los paquetes TS que llevan información privada al respecto del sistema de "scrambling" (datos EMM o ECM)

***Network Information Table**

Esta tabla es opcional y de uso privado. Los paquetes TS que llevan esta información tienen un PID que está definido en la "Program Association Table" al no tener valores específicos. La información queda dividida en secciones denominadas "secciones privadas". Los datos a usar no están definidos por tratarse de datos privados, pero estarán relacionados con

aspectos físicos de la red usada para transmitir el TS.

Secciones Privadas

Además de los datos de la tabla de información de la red, se pueden mandar otra serie de datos privados usando la estructura de una sección privada que se envía en el TS dentro de la categoría de PSI.

La sección privada tiene una estructura análoga a la de las otras secciones para tablas de PSI. Puede así mandarse toda una tabla de datos privados, debidamente identificada y con detección de errores por código CRC.

Sin embargo, existe la opción de usar la sección privada con una estructura más simple, que permite la introducción de más datos pero se pierde protección contra errores o pérdidas de secciones.

Descriptores

Los descriptores usados en el TS son los mismos que los usados en el PS, por lo que el lector puede referirse a las páginas correspondientes.

COMENTARIOS FINALES

La consulta detallada y definitiva de aspectos concretos de la estructura de la señal MPEG-2 sólo puede realizarse en la documentación del estándar. Sin embargo, espero que este libro permita al lector que se inicia, e incluso al ya iniciado en el tema, el conocimiento de aspectos existentes en dicha documentación que permiten comprender la potencia y posibilidades con que se ha dotado a una estructura de señal digital que actualmente está llamada a ser la "banda base" de la TV digital, con aplicaciones que permitan transmisiones multiprograma por cada canal estándar de las bandas de transmisión y difusión actuales.

Los aspectos relativos a la capa de compresión pueden quedar fácilmente claros a cualquier lector, pues se trata simplemente de procesado y organización de datos. Sin embargo, algunos aspectos del "Program Stream" y especialmente del "Transport Stream" pueden no quedar muy claros a quienes no hayan tenido contacto previo con los sistemas de transmisión por paquetes.

Añado dos anexos informativos, no necesariamente exhaustivos, para completar el conocimiento de ciertos aspectos del procesado de la capa de compresión.

El paso siguiente a dar por los interesados en la temática de TV digital es el de analizar los aspectos y normas de aplicaciones de distribución de señal MPEG-2. Concretamente, para difusión, existen las normas DVB ("Digital Video Broadcasting") para aplicación en Europa. Estas cubren los aspectos de difusión por cable, por satélite y por emisores terrestres, así como las instalaciones de recepción.

ANEXO 1: D.C.T. (Transformada Discreta del Coseno)

Según la documentación ISO/IEC 13818, la DCT seguirá las siguientes normas.

La DCT bidimensional NxN se define como:

$$\mathrm{F(u,v)} = \frac{2}{\mathrm{N}} C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$

$$C(u),C(v) = \frac{1}{\sqrt{2}}\ si\ u,v = 0$$

$$C(u),C(v) = 1\ resto\ de\ casos$$

u,v,x,y=0,1,2,3,...,N-1
x,y son coordenadas espaciales
u,v son coordenadas del dominio transformado

La transformada inversa (IDCT) se define como:

$$f(x,y) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$

$$C(u),C(v) = \frac{1}{\sqrt{2}}\ si\ u,v = 0$$

$$C(u),C(v) = 1\ resto\ de\ casos$$

La entrada a la DCT y la salida de la IDCT se representará con 9 bits. Los coeficientes, es decir, la salida de la DCT y la entrada a la IDCT, se representarán con 12 bits.
El margen dinámico de los coeficientes DCT está comprendido entre [-2048:+2047]

La IDCT NxN cumplirá el estándar IEEE 1180-1990 sobre la implementación de IDCT 8x8, en lo relativo a evitar la acumulación de errores..

ANEXO 2: V.L.C. ("Variable Length Coding")

En este anexo se mostrarán algunas tablas de asignación de VLC estandarizadas en MPEG-2. No se trata de un listado exhaustivo, sino un ejemplo para entender el funcionamiento. Las tablas completas pueden obtenerse de la documentación.

La norma especifica códigos VLC no sólo para los valores de los coeficientes DCT, sino también para otros valores que deben transmitirse. Así, se tienen también tablas de definición de VLC para datos sobre macrobloques y vectores de movimiento.

Esta tabla es parte de la "tabla cero" de asignación de VLC a coeficientes DCT.

VLC	run (longitud de ceros)	nivel (±N) (tras los ceros)
1s	0	1
011s	1	1
0100s	0	2
0101s	2	1
00101s	0	3
00111s	3	1
00110s	4	1
10	End of block	
000001	Escape	

* "s" será 0 para nivel positivo y 1 para nivel negativo.
* "End of block" es el código que permite no enviar más datos del bloque si el resto son ceros
*"Escape" seguido de un valor, permite codificar cualquier otro valor [run,nivel] que por su poca probabilidad no aparezca definido en las tablas VLC de la norma. El valor usa 18 bits fijos, 6 para ‘run’ y 12 para ‘level’.

ANEXO 3: DIAGRAMAS GENERALES

Las siguientes páginas muestran varios diagramas generales que pueden servir al lector para aumentar la comprensión de algunos de los aspectos estudiados. Concretamente aparece un diagrama general del proceso de compresión y tres diagramas que considerados consecutivos muestran la secuencia de extracción de tablas y separación de paquetes en un receptor demodulador de MPEG-2.

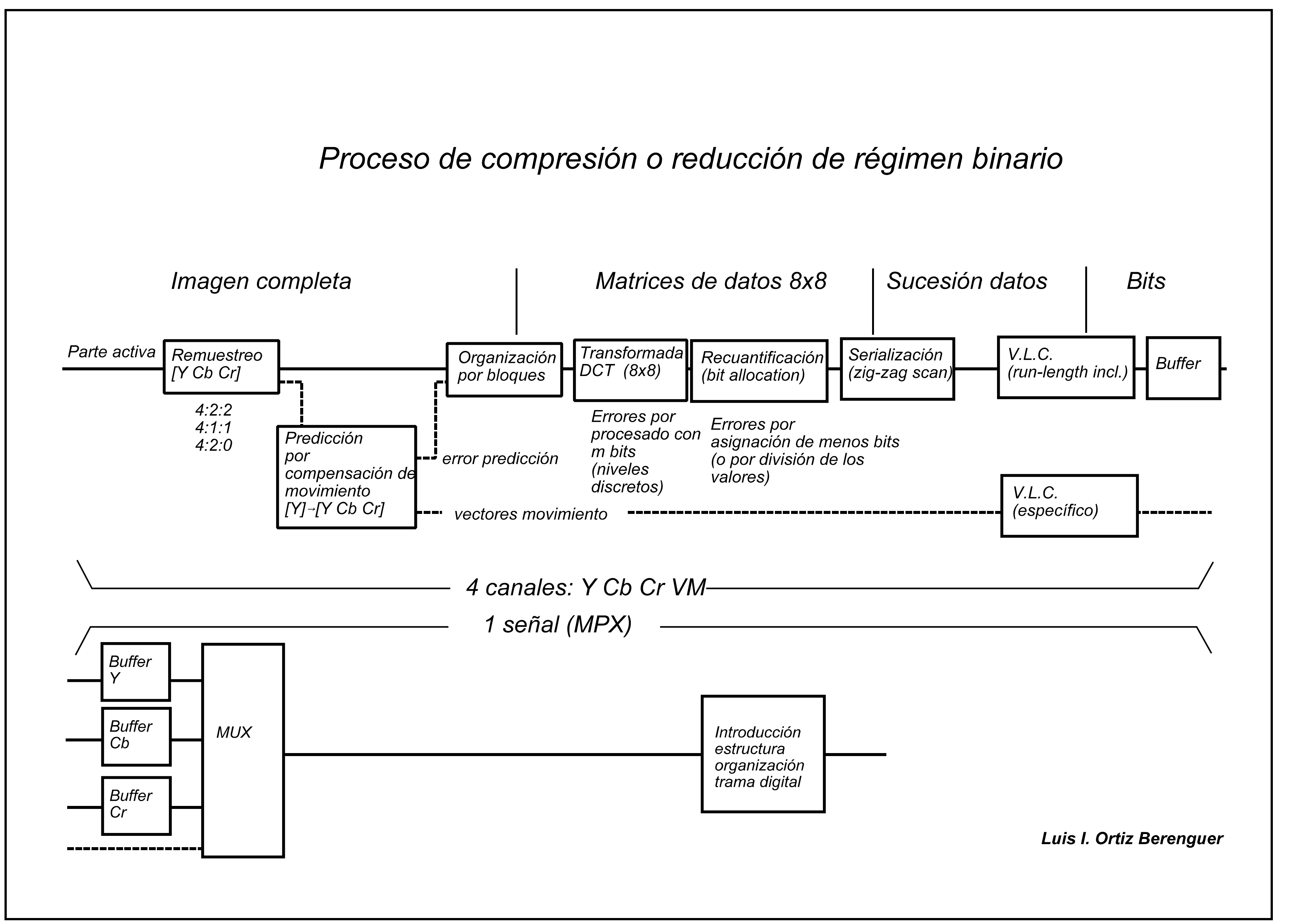


Fig.I-23.- Diagrama general del proceso de compresión

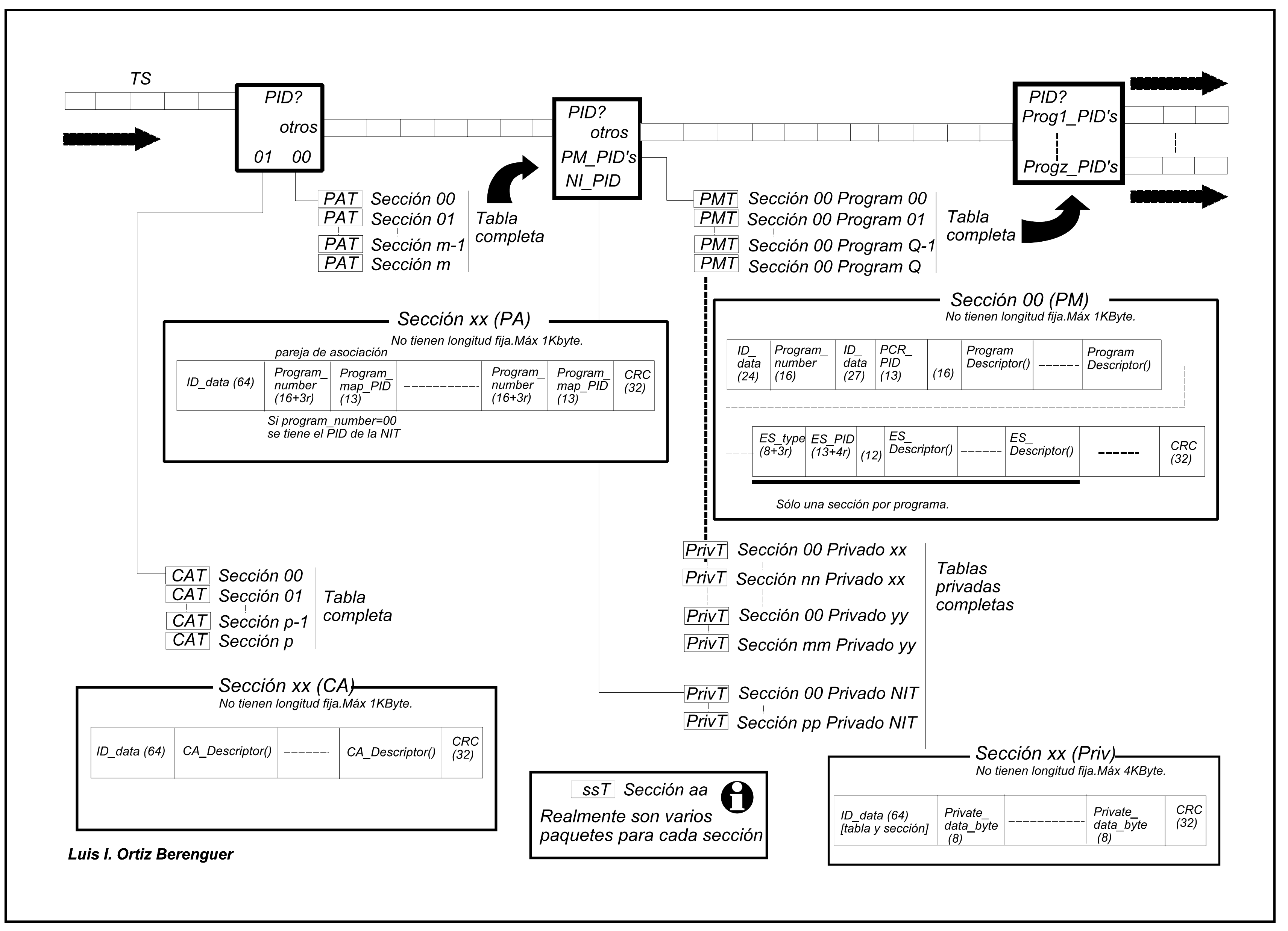


Fig.I-24.- Extracción de los paquetes TS con tablas PSI

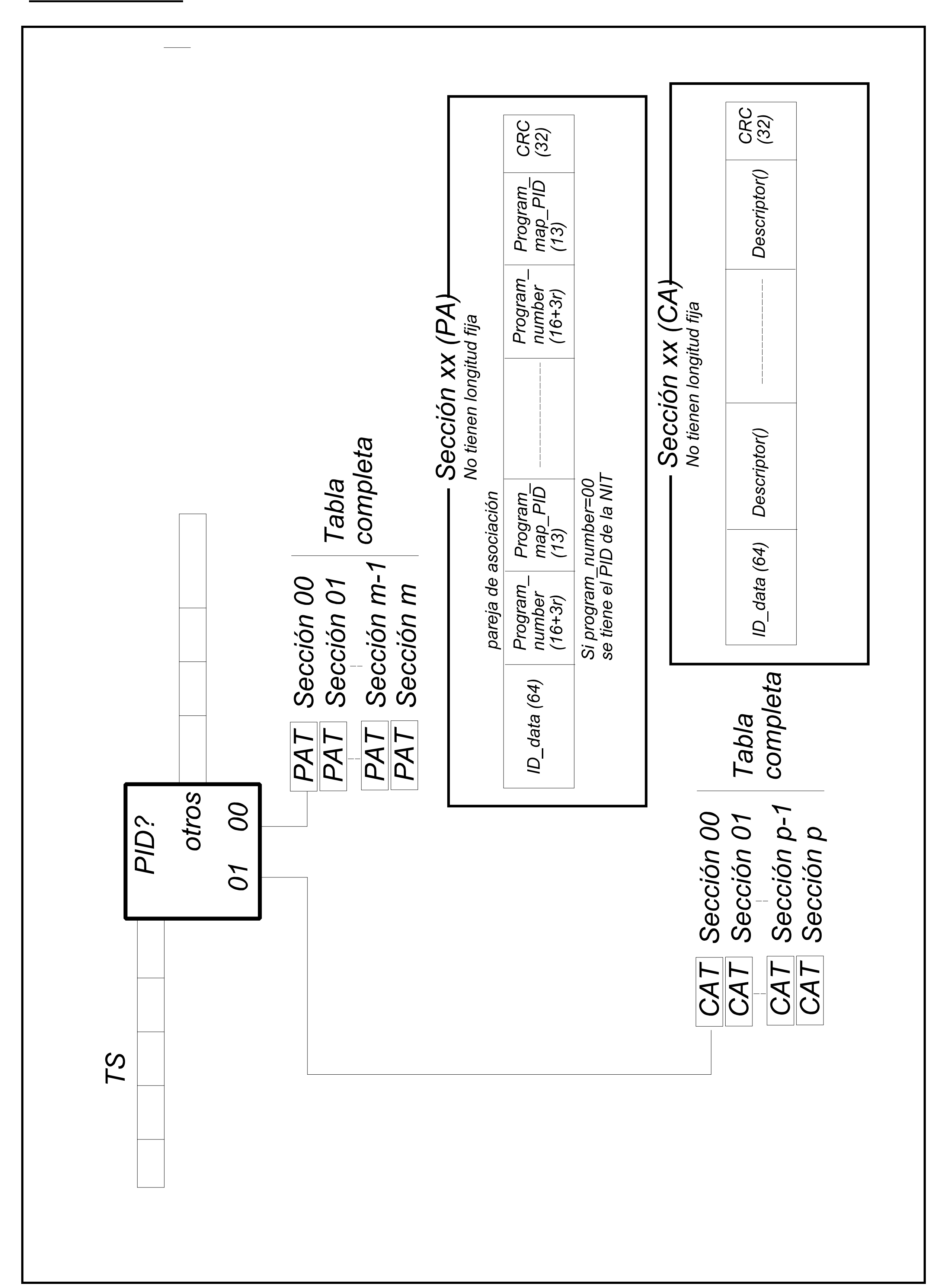


Fig.I-25.- Detalle de extracción de los paquetes del TS con tablas CAT y PAT

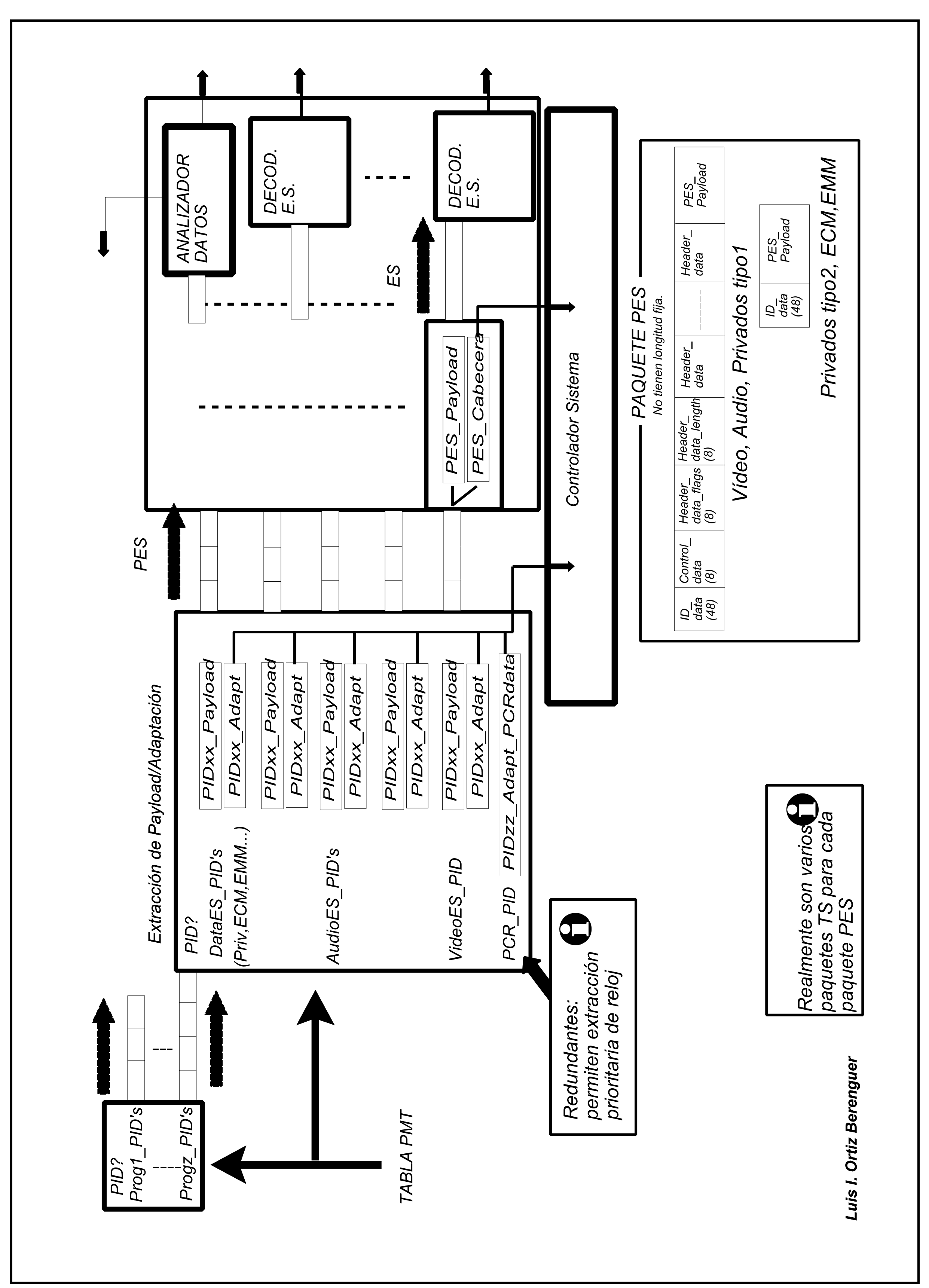
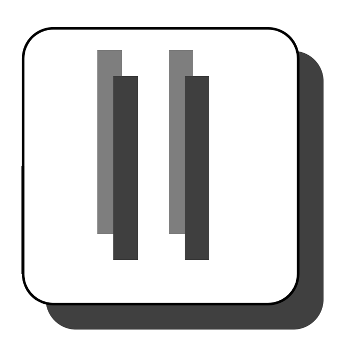


Fig.I-26.- Extracción de los PES



TELEVISIÓN DIGITAL: D.V.B.

INTRODUCCIÓN

La señal digital de estudio es la considerada como de calidad de referencia, aunque hay aplicaciones en las que se manejan señales digitales correspondientes a una mayor calidad de imagen. Las características más significativas de la señal digital de estudio son:

*Señal en componentes
*Muestreo ortogonal y en todas las líneas
*Régimen binario serie:

*SDTV	4/3	270Mbps (4:2:2 con 10 bits)
	16/9	270Mbps (anamórfico)
		360Mbps (4:2:2 extendido)
*HDTV	16/9	1.5Gbps (74.25/37.125 ; 1920x1080)

Lo ideal sería poder disponer en los receptores de dicha señal, sin embargo esto es imposible pues como se ve el régimen binario tan elevado requeriría de grandes anchos de banda en los canales de difusión. Estos canales de difusión siguen siendo los mismos que hasta ahora y sus anchos de banda están estandarizados con los siguientes valores típicos:

*Satélite banda DBS: 27MHz

*Satélite bandas FSS: 27-36MHz

*Terrestre: 8MHz

*TV Cable: 8MHz

De lo indicado anteriormente se deduce la evidente necesidad de reducción del régimen binario para difusión. Esto supone que la señal que tendremos disponible en los receptores no será de la misma alta calidad que la de estudio, sin embargo será una buena solución si la calidad es superior a la disponible hasta ahora con difusión analógica de señal PAL.

Por tanto, a partir de ahora, siempre que se hable de difusión de TV digital se entenderá que la señal banda base es digital comprimida.

DIFUSIÓN DE TELEVISIÓN DIGITAL: D.V.B.

Filosofía:

Algunos de los puntos claves de las normas de TV digital D.V.B. ('Digital Viveo Broadcasting') son:

*Los sistemas permiten transportar combinaciones **flexibles** de vídeo, audio y otros datos para lo cual se hace uso como banda base de la transmisión del múltiplex conocido como MPEG-2 Transport Stream

*Usan un sistema común de información sobre los programas basado en tablas denominado "Service Information"

*Todos usan como procesado inicial un mismo proceso de aleatorización seguido de un mismo FEC tipo RS en un primer nivel ('outer code') lo que permite transmodulaciones sin llegar a decodificar totalmente la señal.

*Las modulaciones y códigos de canal se eligen para adecuarse a cada medio de transmisión.

*También se dispone de un interface común para sistemas de acceso condicional.

*En todos los casos, el criterio de calidad de transmisión que se busca conseguir es el de una tasa de error (BER de la señal banda base) del orden de 10^{-11} o menor, lo que supone menos de 1 error por hora de señal transmitida a una velocidad típica de pocos Mbps del TS-MPEG2. Esta situación de tasa de error se denomina QEF (Quasi-error-free).

Familia DVB

DVB-S: Para difusión por satélite en la banda 11/12GHz. Se puede configurar la señal para varios anchos de banda de los 'transponders' (ETS 300 421)

DVB-C: Para difusión por cable para canales de 8MHz. (ETS 300 429)

DVB-T: Para difusión terrestre en canales de 7 ú 8 MHz.(ETS 300 744)

DVB-MVDS: Para distribución terrestre en banda de microondas por encima de 10GHz.(ETS 300 748)

DVB-CS (DVB-SMATV): Para instalaciones de antena colectiva. Es una adaptación de los anteriores para sistemas (S)MATV.(ETS 300 473)

Otras auxiliares (teletexto, Interface común para acceso condicional, servicio de información,...)

Señales base

La base de la transmisión DVB supone elegir un método de codificación con compresión para las señales, siendo los usados:

Vídeo: MPEG-2

Audio: MPEG Layer II (MUSICAM)

Señal banda base

La banda base que entra en los moduladores DVB son señales de MPEG-2 "Transport Stream", formado a partir de señales base "paquetizadas" (PES de vídeo y audio comprimidas según MPEG-2). Dichos TS pueden contener paquetes de diversos programas y de otros tipos de datos. Así, en un sólo canal de transmisión ocupado por una señal DVB se pueden estar transmitiendo varias informaciones independientes que podrán seleccionarse posteriormente en el receptor-decodificador.

DVB-SATÉLITE

Estructura de datos, código de canal y modulación

El diagrama general del proceso es el de la figura II-1. Los interfaces de entrada/salida del modulador y del demodulador DVB corresponden a señal TS-MPEG2 en el lado de banda base y señal modulada en el otro lado.

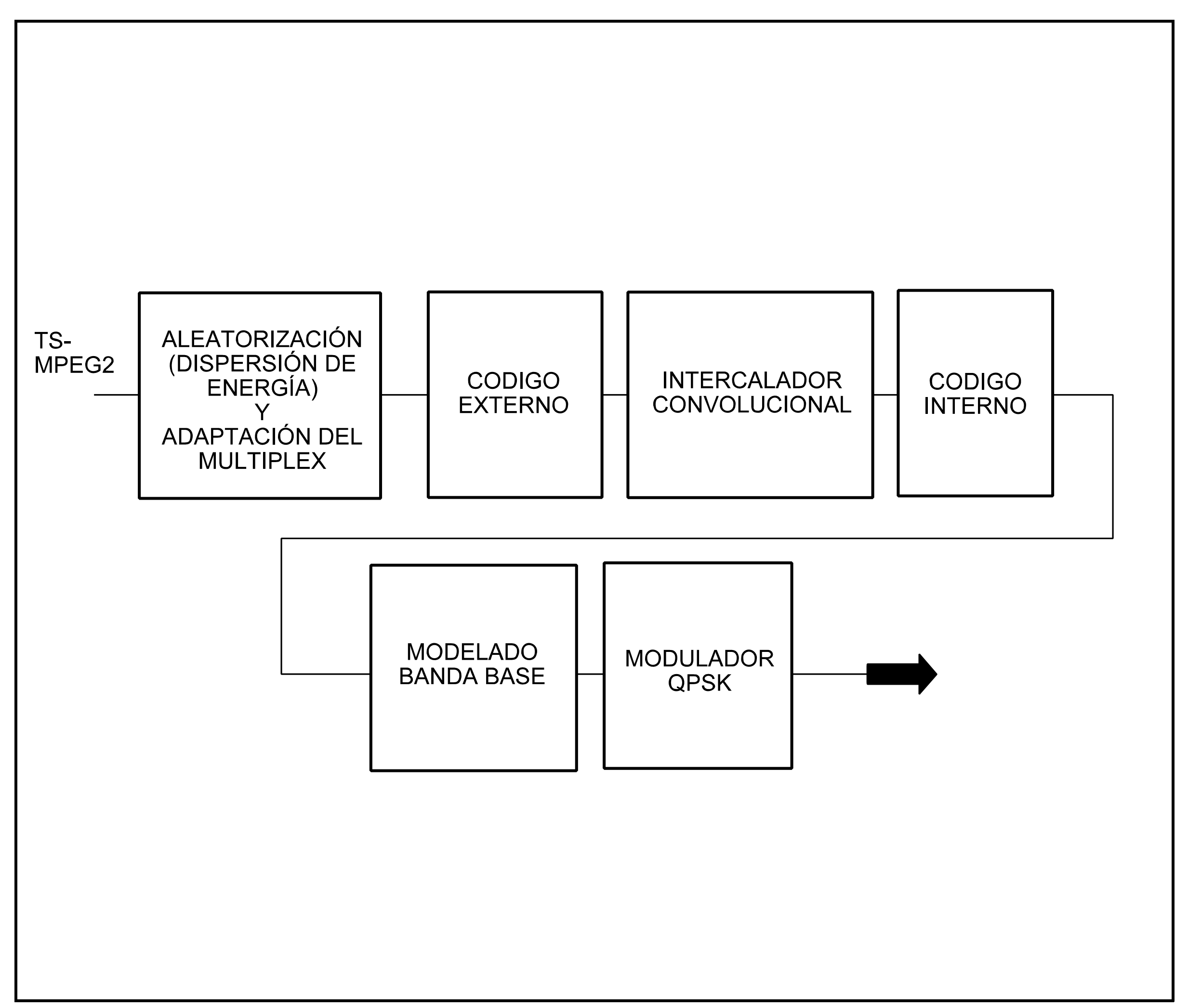


Fig.II-1.- Diagrama general modulador DVB-S

Las transmisión de los bytes siempre se hace empezando por el bit más significativo (MSB)

Adaptación y aleatorización

La señal TS consiste en una sucesión de paquetes de 188bytes cada uno de los cuales tiene un primer byte de sincronismo ('0100 0111' ó 0x47) y luego 3 bytes de resto de cabecera y 184 bytes de datos ('payload' o/y 'campo de adaptación'). De todos ellos, el byte de sincronismo no debe sufrir ningún proceso que lo enmascare o lo haga difícil de identificar. Además, servirá para sincronizar algunos aspectos de la transmisión DVB en el demodulador.

La aleatorización consiste en realizar una suma binaria o módulo-2 (mediante una operación 'or-exclusiva') entre el bit a transmistir y el bit de salida de un registro generador de secuencia pseudoaleatoria (PRBS).

El PRBS se obtiene de un registro de desplazamiento de 15 unidades de memoria de bit que se carga al comienzo del proceso con la palabra de 15bits '1001 0101 0000 000' y que se va realimentando a su entrada con el resultado de la or-exclusiva de las posiciones 14 y 15 del registro. Esa misma señal de reentrada se usa como señal de salida del PRBS.

La denominación matemática usada para definir este proceso es el polinomio $x^{15}+x^{14}+1$

En la figura II-2 aparece un esquema del aleatorizador.

Puede verse que existe una puerta de habilitación para que la secuencia pseudoalatoria no progrese hacia la señal, pero que no desactiva la ejecución de dicha secuencia. Esto permite que el byte de sincronismo de cada paquete se transmita sin aleatorizar, facilitando su identificación.

Además, la secuencia pseudoaleatoria debe ser reinicializada cada cierto tiempo, volviendo a introducir la palabra inicial en los 15 registros de memoria. Esta operación se realiza cada 8 paquetes TS-MPEG2 y tiene lugar al comienzo del noveno paquete. Se puede

decir por tanto que la secuencia se repite con un periodo de 1503 bytes.

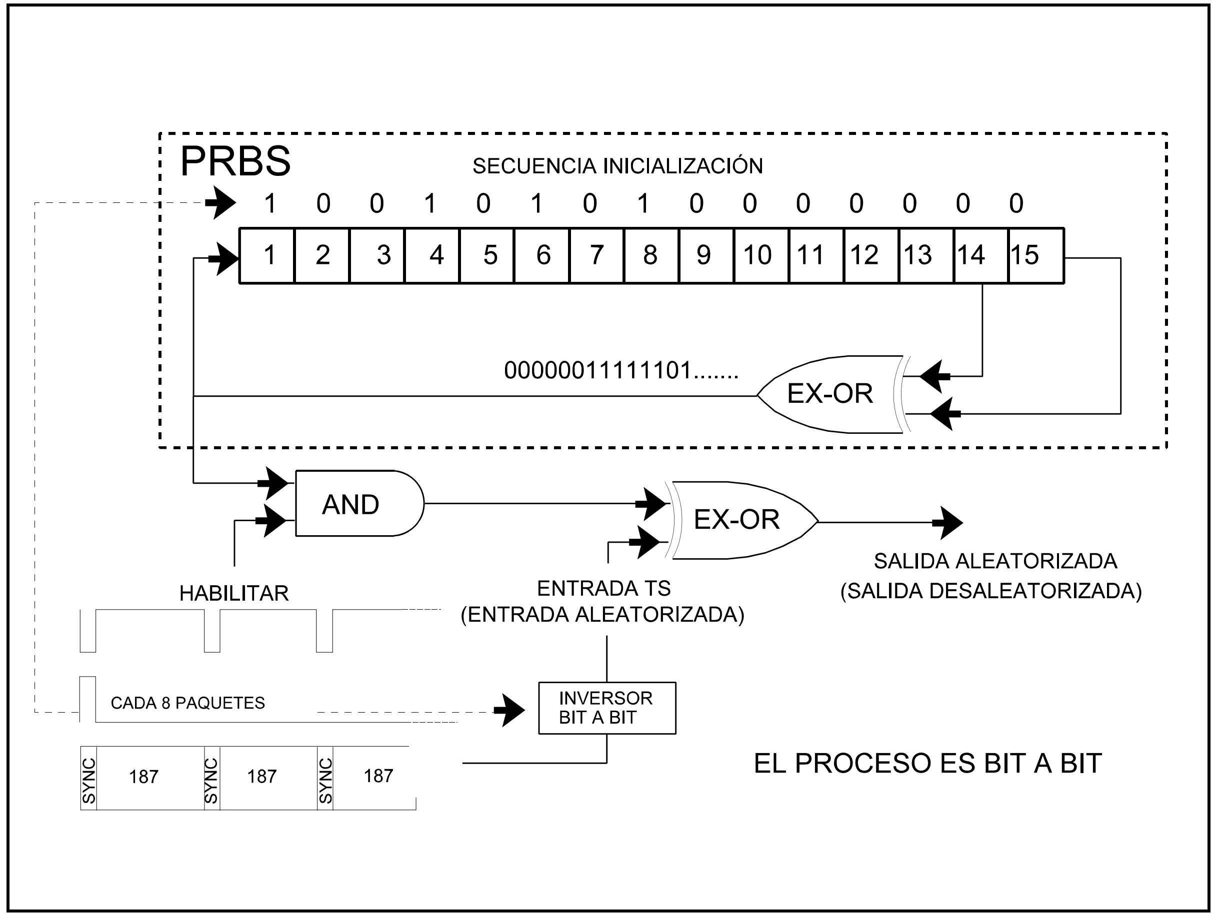


Fig.II-2.- Aleatorizador/Desaleatorizador

Puede comprobarse que 8 paquetes tienen 1504 bytes, por lo que la cuenta anterior parece no salir. La diferencia se debe a que durante el primer byte (el de sincro) de cada 8 paquetes es cuando se iniciliza el registro sin obtenerse salida de él. De hecho, la secuencia pseudoaleatoria se para durante ese byte y se inicia coincidiendo con el primer bit (concretamente MSB) del siguiente byte del paquete. Así queda claro que el periodo es de 1053 bytes.

Puede verse también que la secuencia, a partir de la inicialización, es siempre la misma (es periódica) y por lo tanto, el demodulador lo único que necesita es una referencia precisa del momento en que inicializar su PRBS.

El generador PRBS del demodulador tiene la misma estructura que el del modulador. El circuito es en ambos casos el de la figura II-2 .

Para sincronizar el proceso de inicializar el registro del aleatorizador, el byte de sincro donde se inicializa, se transmite invertido bit a bit (esto es, se transite un 0xB8 en vez de un 0x47).

Por último, indicar que la norma establece que el generador PRBS esté generando la secuencia de forma contínua incluso cuando no haya señal TS, de forma que siempre haya bits que lleguen a modular la portadora y que ésta nunca se emita sin modular. También funcionará del mismo modo en el caso de que los datos de entrada no sean TS-MPEG2.

Código exterior

Como código exterior para corrección de errores (FEC) se utiliza un Reed-Solomon calculado en base a cada paquete y que añade 16 bytes de redundancia permitiendo corregir hasta 8 bytes erróneos. Se aplica también a los bytes de sincronismo. Una de las consecuencias de su aplicación es evidentemente un incremento del régimen binario a transmitir respecto al banda base.

La denominación es RS(204,188), que indica que de cada 188 bytes obtiene 204 bytes. Más concretamente se denomina RS(204,188,t=8) pues corrige hasta 8 palabras. Las palabras son de 8bits (son bytes) y por ello RS opera en un campo de Galois de 2^8 (256) elementos, o dicho de otro modo con todas las posibles palabras de 8 bits como elementos base de las operaciones matemáticas (pero no bit a bit), por lo que el significado de los polinomios no es tan evidente como pudiera parecer.

Además, la generación de códigos RS exige una cierta relación entre el tamaño de los bloques de entrada y salida y el número de palabras a corregir, por lo que en principio no

pueden conseguirse aplicar a cualquier cantidad de palabras de entrada.

Las condiciones para un código RS(n,k,t) son: $n=2^m-1$ y $k=n-2t$, con m=nº de bits de las palabras. Es fácil ver que con m=8 y n=204 no se podría hacer. Hay que calcular un RS con n=255 a partir de k=239 , añadiendo 2t=16 bytes de redundancia para corregir hasta 8 bytes.

Para conseguir un RS(204,188,8) se recurre al denominado 'código acortado', que consiste en añadir 51bytes a cero a la izquierda de los 188 bytes, calcular el RS(255,239,8) y luego eliminan los 51bytes iniciales. La capacidad de corrección no se ve alterada.

La caracterización concreta del código exige indicar a)el campo de Galois usado mediante su elemento primitivo y su polinomio generador del campo, y b) el polinomio generador del código.

Para DVB se tiene:

Elemento primitivo	$\lambda=02_{HEX}$
Polinomio generador del campo	$p(x)=x^8+x^4+x^3+x^2+1$
Polinomio generador de código	$g(x)=(x+\lambda^0)(x+\lambda^1)(x+\lambda^2)....(x+\lambda^{15})$

'Interleave' (Intercalado)

Como proceso complementario en la estrategia de corrección de errores (Forward Error Correction) y para combatir las situaciones de errores consecutivos, denominados largos, se realiza un 'interleave' o intercalado consistente en desperdigar los datos a posiciones localizadas pero no consecutivas. Los errores largos o de ráfaga ('burst') están caracterizados porque existe una relación entre los bits dañados (no son estadísticamente

independientes o aleatorios) consistente en estar dispuestos uno a continuación de otros correspondiéndose sus datos a una misma palabra o palabras consecutivas. Por esa razón los métodos basados en palabras código (p.e. RS) no son suficientemente potentes para corregirlos y además los errores suelen ser mayores que el rango de corrección.

El 'interleave' aleja y 'desordena' los datos consecutivos de forma que, ante la existencia de errores largos consecutivos, al reagrupar el demodulador los bits según el orden original (proceso de 'de-interleave') se conviertan en un montón de errores cortos (pocos bits) aleatoriamente distribuidos (sin relación aparente entre ellos).

De las posibles formas de realizar un 'interleave', DVB elige el intercalado convolucional, en el que la separación entre dos datos originalmente consecutivos no es siempre la misma.

Esto se realiza con una 'batería' de I registros de desplazamiento FIFO (a I se le denomina profundidad del intercalador o número de ramas) cada uno de ellos con una longitud (o número de posiciones de memoria o profundidad del registro) distintos, pero múltiplos enteros consecutivos de un valor M (así se tendrán ramas con retardo jM con j=0,1,2...I-1). El dato básico que se memoriza y se desplaza es 1 byte (no un bit como en otras posibles situaciones). Este esquema se ve en la figura II-3.

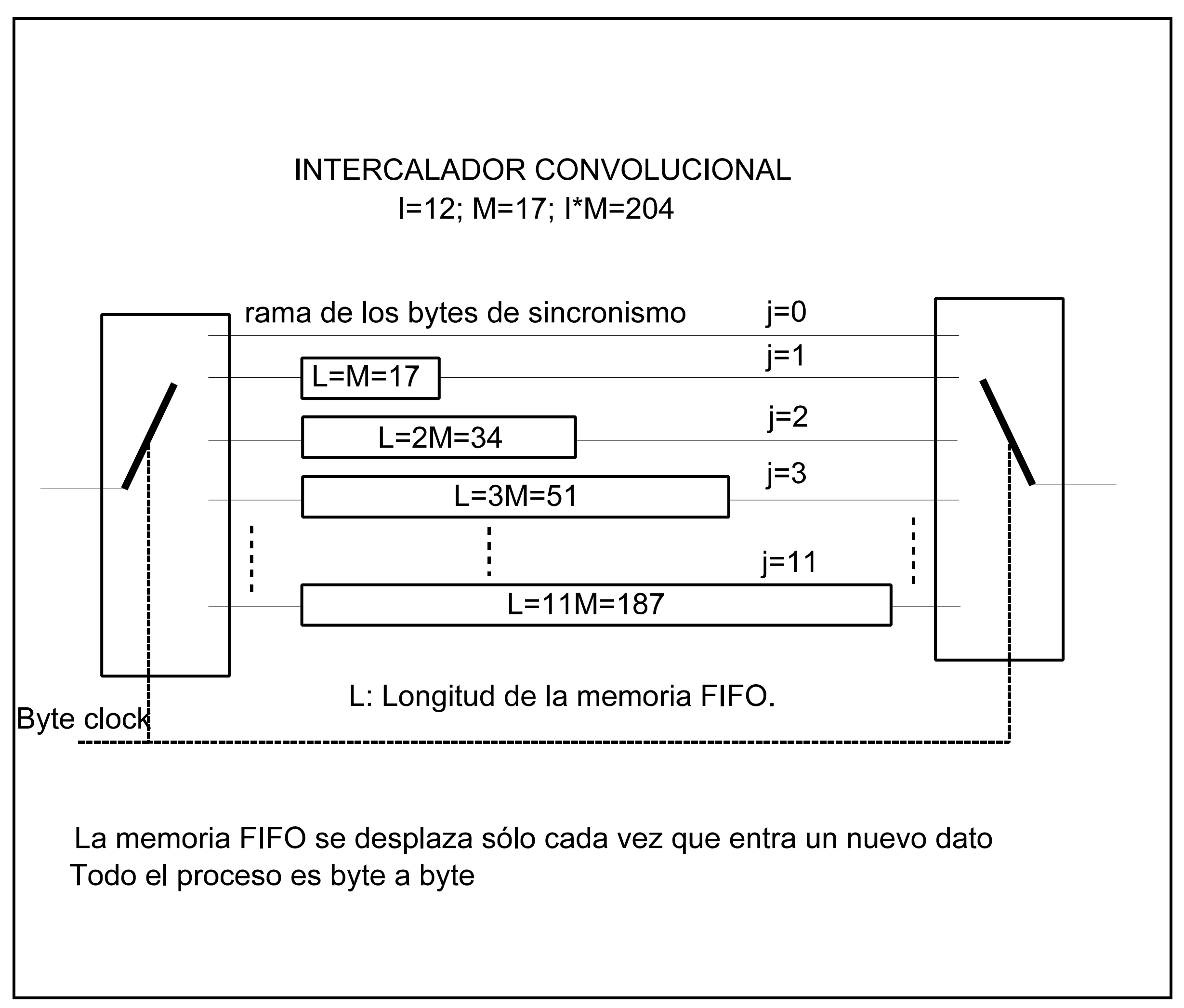


Fig.II-3.- Intercalador convolucional

Al ser registros FIFO, cada desplazamiento dentro de él se realiza al entrar un nuevo dato y no simplemente ante un nuevo ciclo del reloj del sistema.

El valor de I es 12 para DVB y el de M resulta ser 17, pues 12*17=204. De esta manera se puede realizar el intercalado sin afectar a la posición precisa y equiespaciada de los bytes de sincronización, y manteniendo una estructura de paquetes de 204 bytes con el byte de sincro al principio.

Sin embargo, estos paquetes de 204 bytes ya no contienen los datos originales de los paquetes de 188 bytes más los 16 bytes de redundancia, sino que cada paquete intercalado 'n' contiene el byte de sincronismo del paquete original 'n' y otros M-1 (16) datos de dicho

paquete original 'n' en sus posiciones originales, pero escogidos cada I (12) datos. En medio de cada dos valores originales aparecen otros I-1 (11) valores cada uno proveniente de un paquete original distinto pero siempre anteriores, hasta un total de I-1 (11) paquetes anteriores.

En resumen, cada paquete nuevo de 204 bytes se puede dividir en M (17) grupos de I (12) datos. Cada uno de los 12 proveniente de un paquete anterior distinto, empezando por el dato del propio paquete y siguiendo por el dato del paquete más cercano. Cada dato en el nuevo paquete queda en la misma posición relativa al nuevo byte de sincro que la que tenía en el paquete original.

Desde el punto de vista del proceso de intercalado, los datos de cada paquete se desperdigan por I paquetes (el propio y los I-1 siguientes) a razón de M datos en cada paquete. Los datos ocuparán en el nuevo paquete la misma posición relativa respecto al byte de sincro que en el paquete original.

Una forma sencilla de calcular el nuevo orden, consiste en numerar las posiciones originales de los bytes desde 1 y sin limite por la longitud del paquete, y usar ese número de orden para identificar la posición en los paquetes intercalados. Se empieza por el 1 y se dejan I-1 (11) huecos y se pone el siguiente valor no cambiado que será el 13 y se vuelven a dejar I-1 huecos y así sucesivamente se obtienen los datos que no cambian. Se puede comprobar que el dato 205 (el siguiente byte de sincro) queda sin cambiar. Si se quiere comprobar todo deberán listarse los 12 paquetes (12*204 bytes) que se involucran en el intercalado de un paquete original. Después, los valores intercalados entre los no modificados se pueden calcular fácilmente restando del anterior (empezando por el no cambiado) la cantidad I*M-1 (203). El resultado obtenido marca el nuevo orden de los datos originales. Si la cantidad de la resta es negativa, indica que corresponde a datos anteriores al considerado como 1. La 'ristra' de valores obtenidos debe dividirse en paquetes de 204 bytes y se comprobará que el inicial de cada paquete (byte de sincro) es el inicial del paquete original.

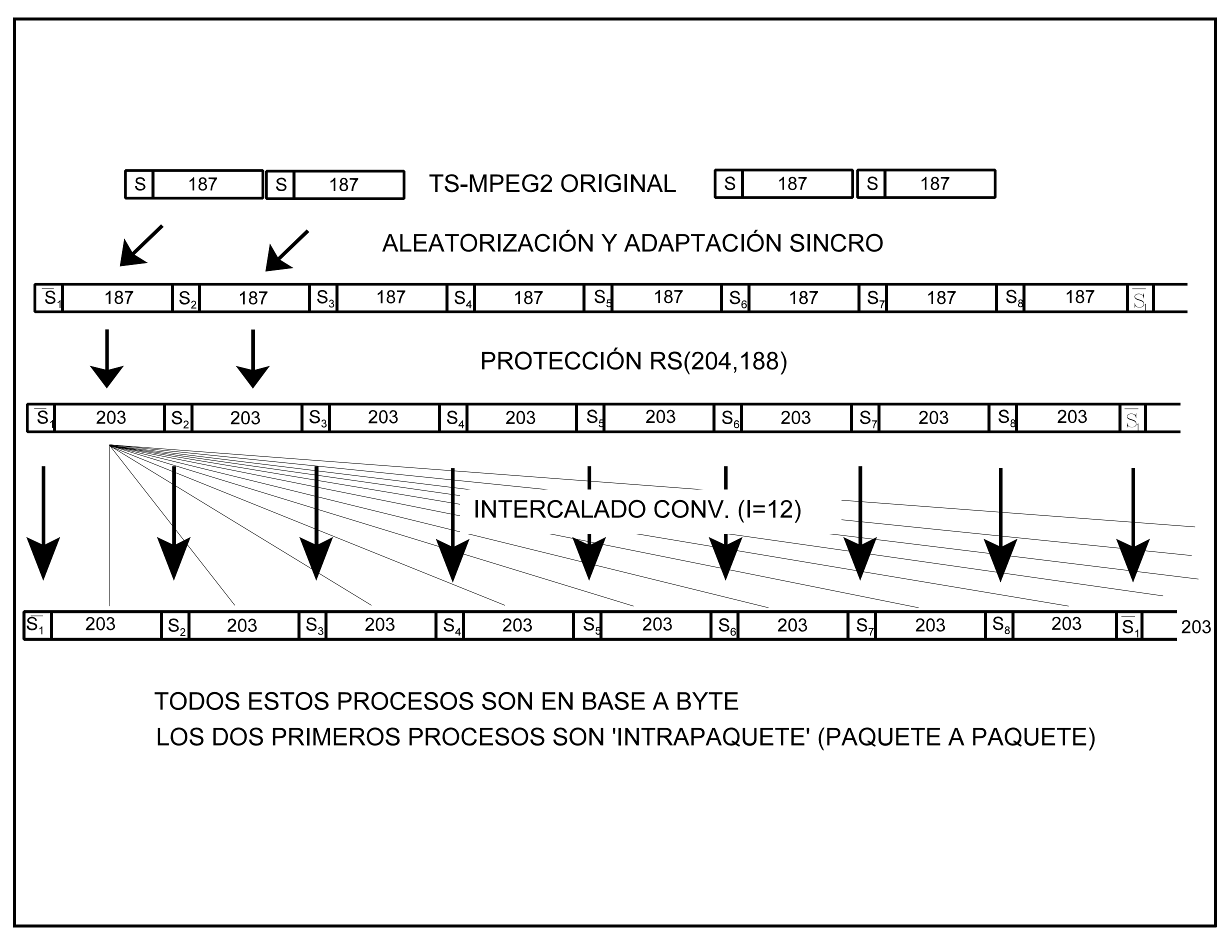


Fig.II-4.- Organización de datos ('cuadro de datos') hasta el intercalador

El lector interesado en hacer una prueba puede intentarlo con paquetes de 12bytes usando un intercalador de I=4 y M=3, que da un resultado manejable y suficientemente significativo con unos 50 valores.

No debe perderse de vista que los bytes de redundancia también han sido desperdigados.

A la estructura de paquetes intercalados se le denomina 'frame' o cuadro. Su naturaleza convolucional no permite establecer un diagrama del proceso de ordenación que parezca un cuadro, como sí hubiera ocurrido en un intercalado por bloques, de donde proviene la denominación de 'frame' que se mantiene en el caso convolucional.

La reorganización de los datos en el demodulador exige el uso de un circuito

parecido pero con la distribución de retardos contraria, como puede verse en la figura II-5.

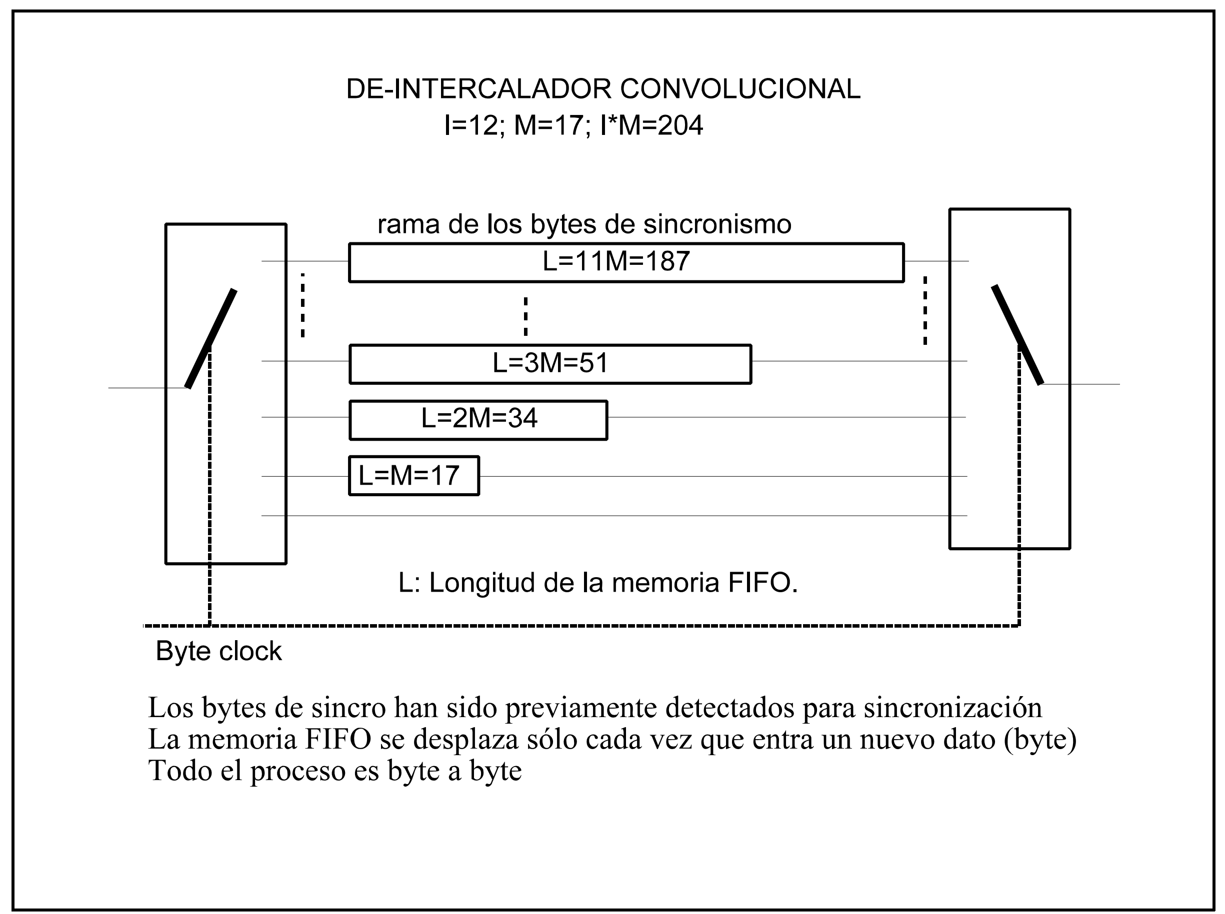


Fig.II-5.- De-intercalador

Código interior

Si la estrategia FEC quedara limitada a los dos procesos anteriores, quedaría expuesta a un resultado malo si existiesen errores aleatorios cortos en la estructura de paquetes intercalados. Por ello es esencial la existencia de un código de corrección de errores cortos aplicado a los paquetes intercalados. Según el tipo de transmisión a realizar, puede interesar que dicho código de corrección también realice funciones de código de canal. Es el caso de satélite, y la norma define un código interior tipo convolucional. La capacidad de corrección y la adaptación al canal son factores que se pueden elegir, aunque establecen un compromiso

al ir uno en contra del otro para un canal tipo satélite.

Un código convolucional funciona en base a bit (no en base a bytes) y obtiene a la salida más bits que los que entran, incrementando el régimen binario. El cálculo de los bits de redundancia se realiza mediante unos registros de desplazamientos y unas operaciones or-exclusiva entre ciertas posiciones de dichos registros. Así, para el mismo bit de entrada, las salidas no siempre son iguales sino que dependen de los bits anteriores, pues el sistema tiene memoria. El funcionamiento suele analizarse agrupando 'n' bits de entrada para obtener 'm' bits de salida, consiguiéndose relaciones no enteras.

La definición de un código convolucional requiere indicar:

Rate	Relación bits entrada/bits salida
Constrain Length 'K'	Número de registros de memoria incluyendo el bit entrante. Se implementan K-1 retardos 2^{K-1} es el número de estados del codificador
Polinomios de código '$G_i(x)$' (uno por cada salida del codificador)	Polinomios que indican qué registros de memoria usar para obtener cada una de las salidas mediante 'or-exclusivas'

DVB fija el valor K=7 en todos los casos y tan sólo dos polinomios, sin embargo se pueden elegir hasta 5 'rates' distintos (1/2, 2/3, 3/4, 5/6, 7/8). Para poder acometer tantas posibilidades con tan sólo dos polinomios se recurre a la asignación 'punctured' a partir de un codificador convolucional base de relación 1/2 con dos polinomios.

Los dos polinomios se representan de forma reducida por un número binario que tiene '1' en las posiciones de las potencias de x que existen en el polinomio (registros a sumar módulo-2) y '0' en las posiciones de las potencias que no existen. Dicho número binario expresado en octal es el dato de la norma. Así DVB asigna los polinomios 171_{oct} y 133_{oct} para obtener las salidas X e Y respectivamente a partir de los bits de entrada.

El codificador base 1/2 corresponde por tanto al esquema de la figura II-6

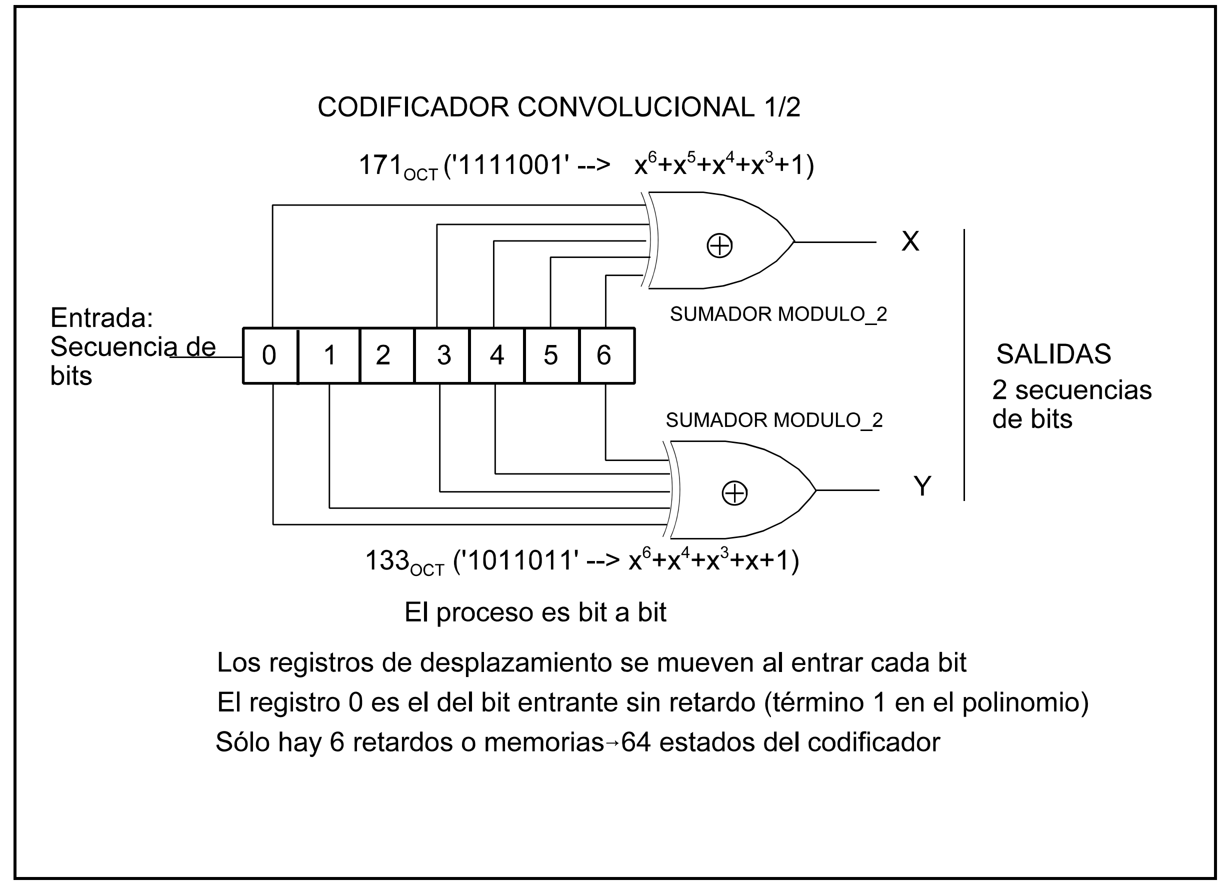


Fig.II-6.- Codificador convolucional 1/2

Además, dado que las modulaciones usadas modifican la fase de la portadora, la señal digital se divide en dos señales denominadas I y Q para atacar al modulador.

El proceso 'punctured' asigna algunos de los datos XY obtenidos como datos IQ definitivos de la codificación convolucional. Para el caso 1/2, cada bit de entrada da lugar a dos bits (uno en señal I y otro en Q). La asignación es directa: I=X; Q=Y para todos los bits de entrada.

Para otros 'rates' m/n, hay que dejar entrar los m bits de entrada para obtener 2m bits de salida en XY y seleccionar sólo n bits de ellos, con un orden concreto, como salidas IQ.

La figura II-7 muestra el esquema conceptual incluyendo los pasos siguientes que

aún no se describen.

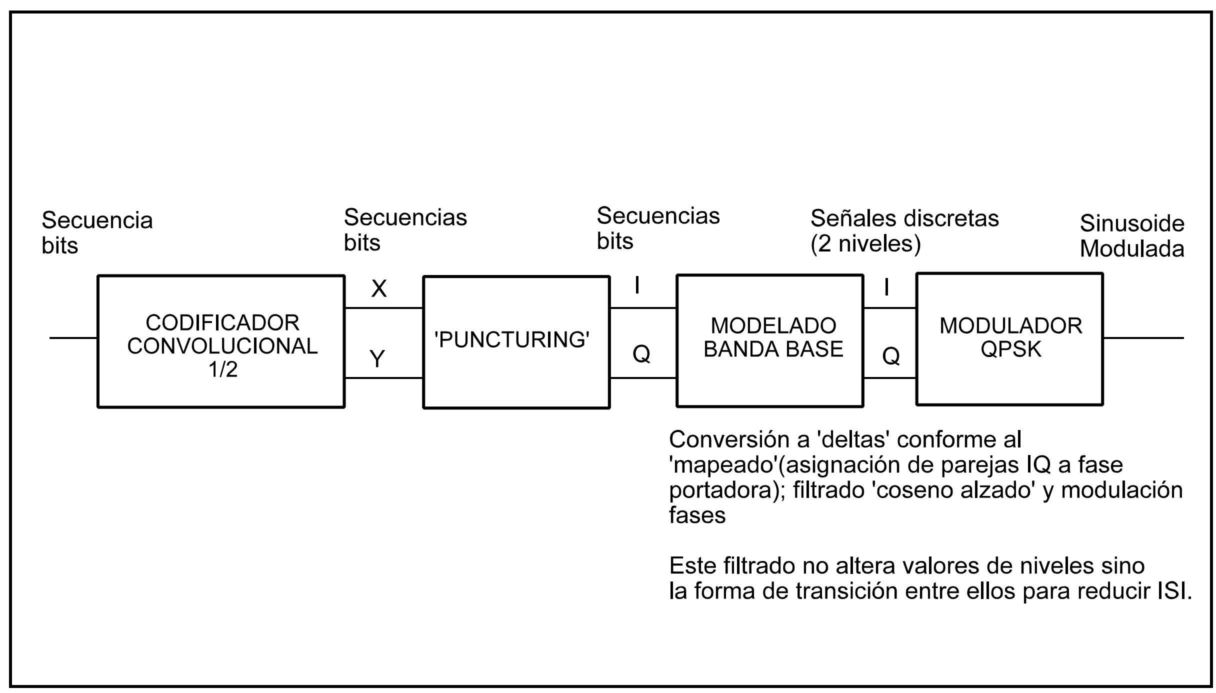


Fig.II-7.- Esquema de codigo interno y modulación

Como ejemplo tomemos el caso de r=3/4

bit entrada	1	2	3	4	5	6	7	8	9
Salidas									
X	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9
Y	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9
Patrón de selección ('puncturing')									
X	1	0	1	1	0	1	1	0	1
Y	1	1	0	1	1	0	1	1	0
Salidas									
I	X_1	Y_2	X_4	Y_5	X_7	Y_8			
Q	Y_1	X_3	Y_4	X_6	Y_7	X_9			

El patrón de selección indica qué datos XY se toman para formar IQ ('1' se toma, '0'

se elimina). El orden es asignando alternativamente valores a I y Q siguiendo el orden de X y luego Y. Se ve que tanto datos de X como de Y aparecen en I y Q.

Hay un cambio en el régimen binario de I y Q respecto al de bits de entrada que parece una disminución, pero no olvidemos que ahora son dos señales y no una, por lo que el régimen binario total es mayor.

Véase que de 6 posibles se toman sólo 4. En el mismo tiempo, por 3 bits de entrada se tienen 4 de salida IQ, es decir un 'rate' 3/4.

La tabla siguiente muestra los valores de la norma DVB para satélite

Codificador Convolucional												
Codificador base			Relación del código ('rate')									
			1/2		**2/3**		**3/4**		**5/6**		**7/8**	
K	G_1 (X)	G_2 (Y)	Punct	d_{free}	Punct	d_{free}	Punct	d_{free}	Punct	d_{free}	Punct	d_{free}
7	171 oct.	133 oct.	X:1 Y:1	10	X:10 Y:11	6	X:101 Y:110	5	X:10101 Y:11010	4	X:1000101 Y:1111010	3
			$I=X_1$ $Q=Y_1$		$I=X_1Y_2Y_3$ $Q=Y_1X_3Y_4$		$I=X_1Y_2$ $Q=Y_1X_3$		$I=X_1Y_2Y_4$ $Q=Y_1X_3X_5$		$I=X_1Y_2Y_4Y_6$ $Q=Y_1Y_3X_5X_7$	

El parámetro 'd_{free}' (distancia libre) evalúa la capacidad de corrección de errores de bit del código convolucional. El análisis no es exacto debido a la naturaleza convolucional del proceso, pero puede decirse que el número de errores corregibles en una secuencia de 21 a 35 bits serán:

$$t = entero\left[\frac{d_{free} - 1}{2}\right]$$

Modulación y modelado banda base

Los bits que componen las señales I y Q sirven para atacar al modulador QPSK en forma de componentes en fase y cuadratura de un modulador QAM. La asignación de cada una de las 4 fases se corresponde a una pareja IQ para cada fase. Puede verse en la figura II-8 el diagrama de constelación del modulador según la norma DVB.

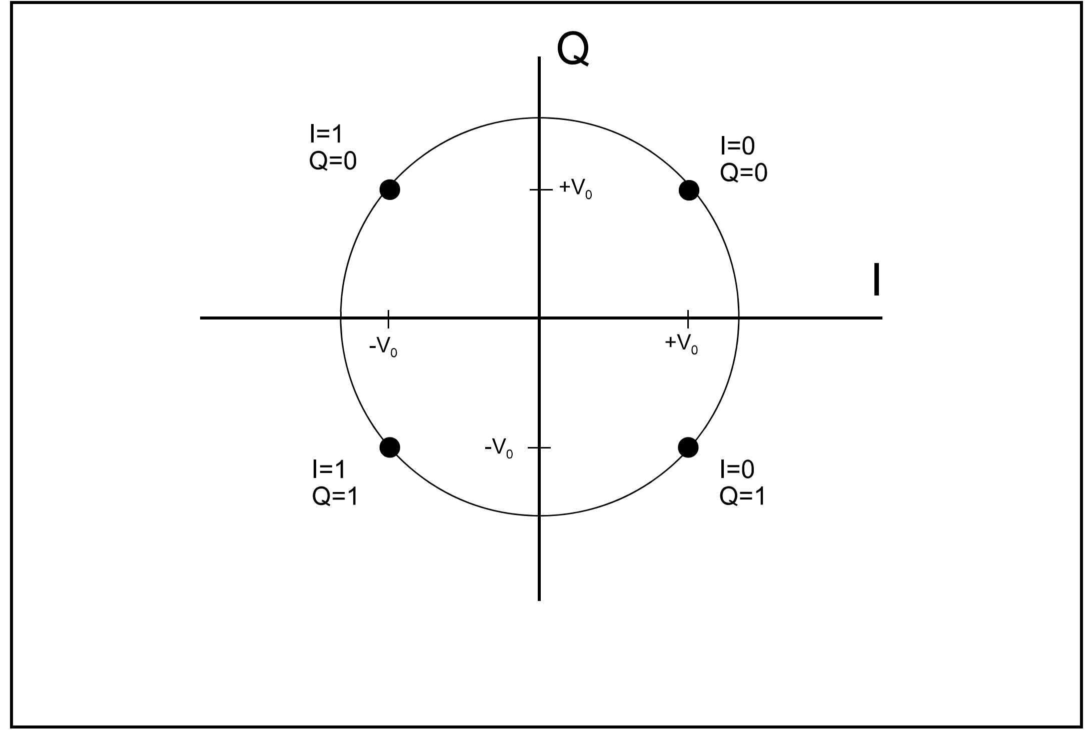


Fig.II-8.- Modulación QPSK

La modulación supone obtener una señal analógica sinusoidal de frecuencia concreta y cambiar su fase entre alguno de los cuatro valores. Cada una de esas cuatro posibilidades se denomina 'símbolo'. Este proceso se realiza más cómodamente mediante una modulación de amplitud en cuadratura. La componente I (modulador en fase) tendrá dos posibles valores y la componente Q (modulador en cuadratura) otros dos. El resultado es una QPSK. Puede verse que los valores de I y Q reales tienen que ser $\pm V_0$ para dotar a la portadora modulada de una amplitud constante de $1.41*V_0$. Esto indica que las señales IQ que entran al modulador no son electrónicamente las mismas IQ que salían del codificador convolucional. debe realizarse una conversión binario digital- binario analógico, obteniéndose no una señal digital

(binaria '0' y '1') sino una señal discreta (matemáticamente deltas con nivel $\pm V_0$). El nivel V_0 puede normalizarse a nivel 1, pero no confundir con '1' valor de bit.

La asignación de la conversión es:

	'0'	'1'
I	+1	-1
Q	+1	-1

Con esta asignación, las parejas IQ digitales son asignadas ('mapeadas') a fases de manera absoluta y con distribución (codificación) Gray, lo que minimiza el error en caso de demodulación defectuosa de fase, ya que de cualquier símbolo (fase) al siguiente cercano sólo hay un bit de cambio.

La existencia de IQ como señales discretas tanto en el modulador como en el demodulador, plantea la problemática de la interferencia entre símbolos que aparece cuando las señales digitales toman formas de onda electrónicas y deben analizarse para identificar si se tiene '1' ó '0'. Para minimizar la interferencia entre símbolos en las señales IQ banda base (no en forma de portadora modulada) se introduce un filtro que modele el canal banda base con una respuesta en frecuencia tipo 'coseno alzado'. Este tipo de respuesta se caracteriza por un parámetro denominado 'roll-off (∝)'. DVB para satélite establece este parámetro en ∝=0.35.

La norma establece una plantilla concreta para la respuesta en frecuencia del filtro de coseno alzado tanto en módulo como en retardo de grupo. En este último, el rizado del retardo de grupo no debe superar $\pm 0.07\ T_S$, en la banda de paso hasta f_N . Siendo $f_N = R_S/2$: Frecuencia de Nyquist igual a la mitad del régimen binario de I ó Q (que coincide con el régimen de símbolo)

El lector puede referirse a textos sobre comunicaciones digitales si quiere profundizar en el filtrado de coseno alzado. En el apartado de DVB-cable se volverá a incidir en el tema

con la perspectiva correpondiente.

Capacidad de transmisión

En la tabla siguiente aparecen los cálculos de capacidad de transmisión para una situación concreta, aunque suficientemente típica. El ancho de banda del transpondedor (canal) de satélite debería, en principio, ajustarse al impuesto por el filtrado de coseno alzado de forma que la relación entre ancho de banda y régimen de símbolos transmitible $BW/R_S=1.35$ (dado que $\alpha=0.35$). Sin embargo, puede aprovecharse un poco más y bajarse esa relación teniendo más R_S para el mismo ancho de banda BW. Eso supone una degradación en la probabilidad de error que se suele compensar exigiendo mayor nivel de recepción.

Régimen binario (Mbps) de señal MPEG-2 para transpondedor con $BW/R_S=1.28$ para varios 'rate' del codificador convolucional							
BW	**R_S**	**$R_{b\,DVB}$**	**1/2**	**2/3**	**3/4**	**5/6**	**7/8**
27MHz	21.1Mbaudio	42.2Mbps	19.4	25.9	29.2	32.4	34.0
33MHz	25.8Mbaudio	51.6Mbps	23.8	31.7	35.6	39.6	41.6
36MHz	28.1Mbaudio	56.2Mbps	25.9	34.6	38.9	43.2	45.4

Condiciones de recepción

La recepción de señal DVB (ver figura II-9) debe hacerse con una E_b/N_o que permita asegurar una tasa de error a la salida del decodificador de Viterbi del orden de $2x10^{-4}$ para que tras el decodificador Reed Solomon la situación sea de QEF (menor de 10^{-11}) del orden de un error sin corregir por hora para la señal MPEG2. La tabla siguiente muestra dichos valores. Están calculados para canal gausiano con el ancho de banda dado por el filtro de coseno alzado y considerando un margen de implementación del receptor de 0.8dB, un

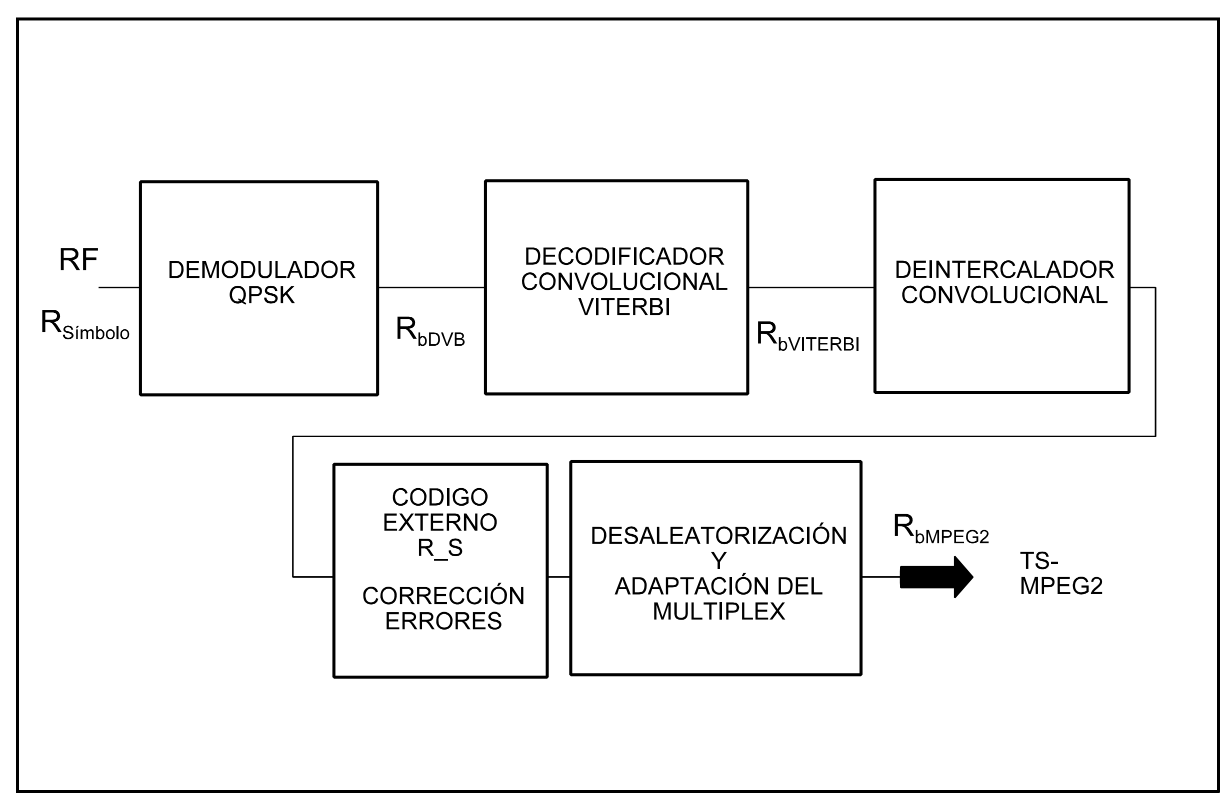


Fig.II-9.- Esquema de receptor DVB-S y regímenes binarios usados

incremento del ancho de banda de ruido debido al uso de código exterior (RS204:188) de 0.36dB y el valor E_b está referido a la energía por bit de señal MPEG2 recuperado. Lo anterior quiere decir que si se quiere calcular la relación C/N necesaria, basta con sumar $10x\text{Log}(R_{bMPEG2} / BW_{RF})$ al valor de la tabla.

E_b/N_o necesaria para QEF en señal MPEG2			
'Rate' convolucional	E_b/N_o (dB)	'Rate' convolucional	E_b/N_o (dB)
1/2	4.5	5/6	6.0
2/3	5.0	7/8	6.4
3/4	5.5		

Como ejemplo concreto, la tabla siguiente resume los cálculos para un transpondedor de 33MHz con una relación de BW/R_S=1.28.

Datos de recepción para transpondedor de BW_{RF}=33MHz y relación BW/R_S=1.28							
$R_{Símbolo}$ MBaudios	R_{bDVB} Mbps	'Rate' conv.	$R_{bVITERBI}$ Mbps	R-S rate	R_{bMPEG2} Mbps	E_b/N_o dB	C/N dB
25.776	51.552	1/2	25.776	188/204	23.754	4.5	4.1
25.776	51.552	2/3	34.368	188/204	31.672	5.0	5.8
25.776	51.552	3/4	38.664	188/204	35.631	5.5	6.8
25.776	51.552	5/6	42.960	188/204	39.590	6.0	7.8
25.776	51.552	7/8	45.108	188/204	41.570	6.4	8.4

El lector que realice los cálculos verá que no le coinciden. Eso se debe a que los valores de C/N de la tabla incluyen unos márgenes que compensan los efectos de los filtros y amplificadores del satélite (+1dB). Además incluye el margen de implementación del receptor(0.8dB) considerado ya en el valor de E_b/N_o utilizado.

$$\frac{C}{N} = \left[\frac{E_b}{N_0}\right]_{teor} + M_{implem} + 10\log\frac{R_{bMPEG2}}{BW_{RF}} + M_{sat}$$

$$con \quad \left[\frac{E_b}{N_0}\right]_{teor} + M_{implem} = \left[\frac{E_b}{N_0}\right]_{tabla}$$

Por último, en caso de disminuir el ancho de banda del canal más respecto al impuesto por el filtro de coseno alzado, es decir, para relaciones BW/R_S menores de 1.35, la figura II-10 muestra la corrección a realizar en el valor teórico de E_b/N_o usado en tablas anteriores.

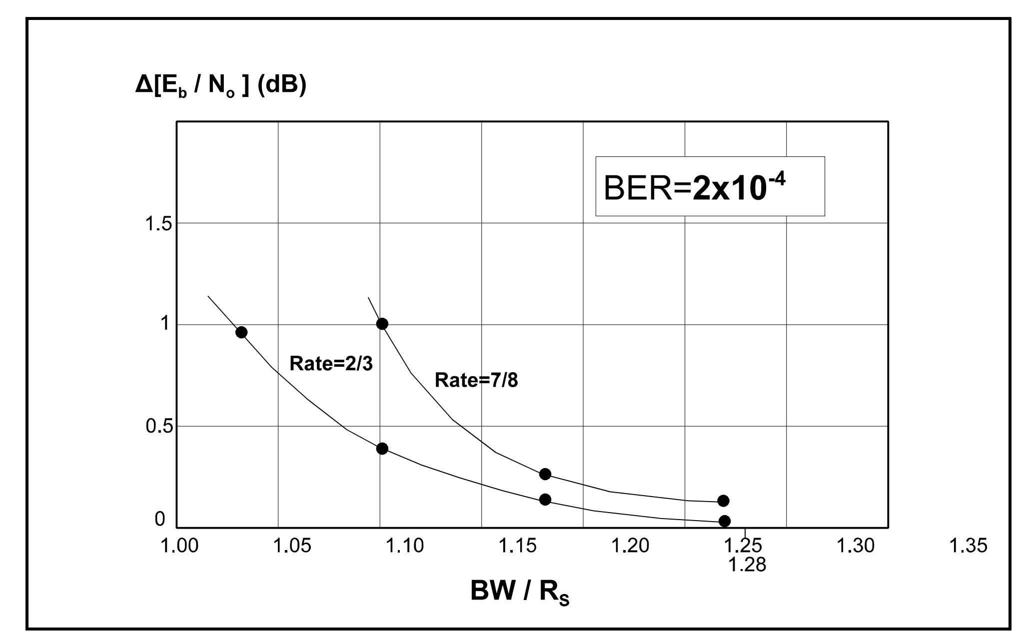


Fig.II-10.- Corrección de E_b/N_o para varios BW/R_S

Como puede verse, no hay apenas corrección entre los casos 1.35 y 1.28, y por eso se usa este último en los cálculos y en algunos casos prácticos en los que se consigue un 5% más de régimen binario para un mismo transpondedor y sin cambio en las condiciones de recepción.

DVB-CABLE

Estructura de datos, código de canal y modulación

Para asegurar la mayor compatibilidad posible y facilitar la transmodulación entre unas señales y otras, las normas DVB tienen en común algunos de los procesos descritos, por lo que no volverán a describirse. Concretamente son comunes a todos los sistemas:

-Adaptación del múltiplex TS y Aleatorización.

-Codificación exterior RS(204,188,t=8)

-Intercalado ('Interleave') Convolucional con profundidad I=11

Existen sin embargo otros procesos que son específicos para cada tipo de transmisión, concretamente los relacionados con el código interior, modelado banda base y modulación. También lo son los correspondientes aspectos de demodulación, detección y

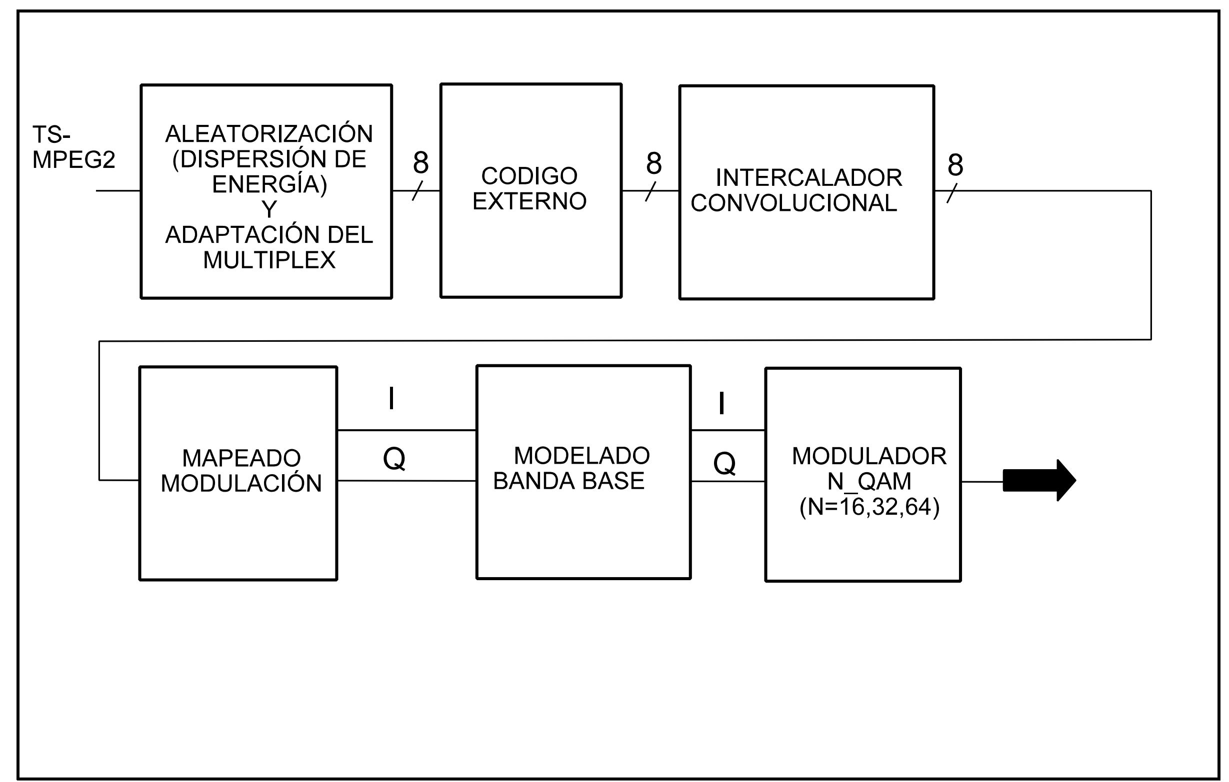


Fig.II-11.- Diagrama general modulador DVB-C

posible ecualización.

El diagrama general del proceso es el de la figura II-11. Los interfaces de entrada/salida del modulador y del demodulador DVB corresponden a señal TS-MPEG2 en el lado de banda base y señal modulada en el otro lado.

Las transmisión de los bytes siempre se hace empezando por el bit más significativo (MSB)

En la figura II-12 se muestra un diagrama del receptor para DVB-C que propone la norma

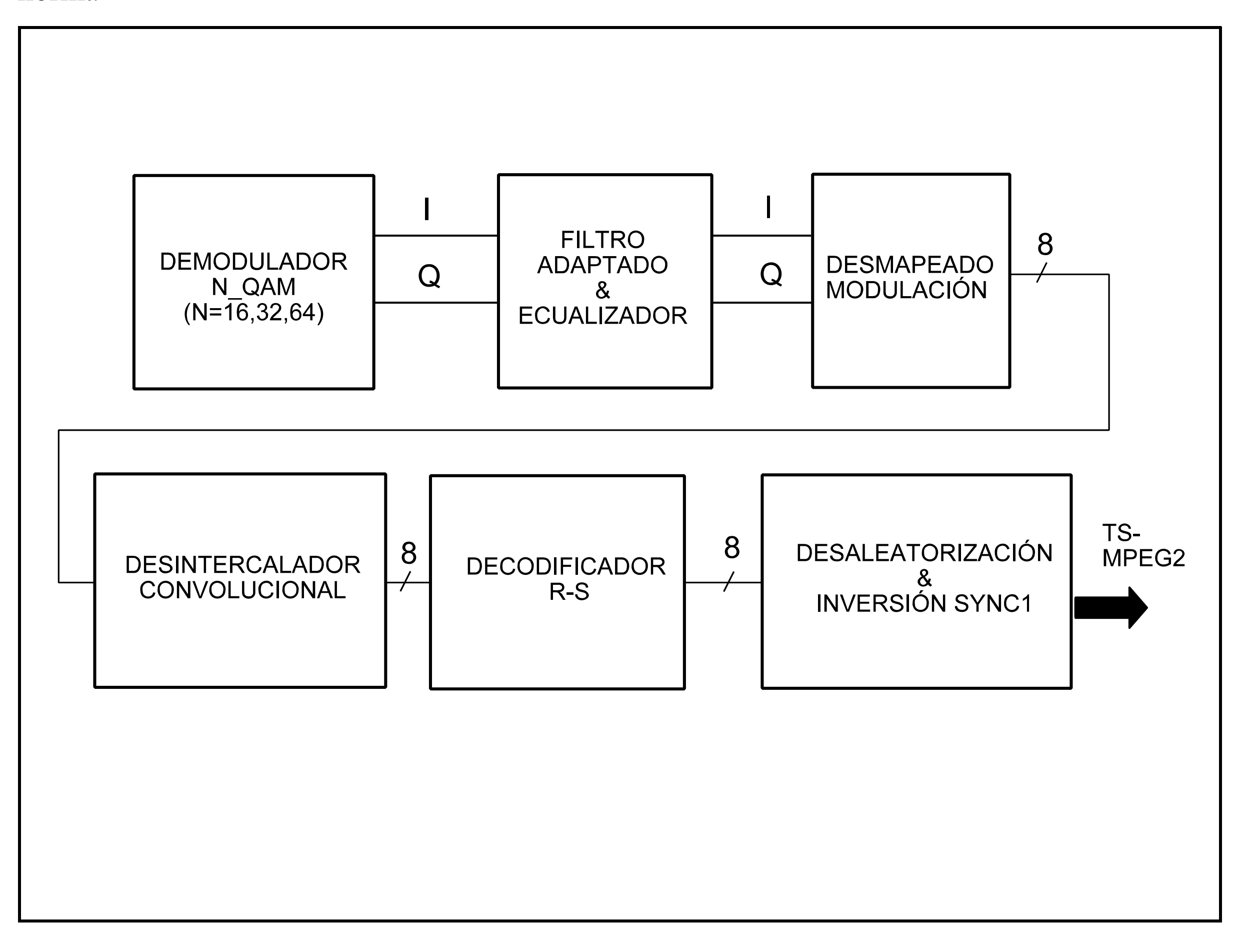


Fig.II-12.- Diagrama general receptor DVB-C

Modulación y modelado banda base

Al contrario que la estructura FEC usada en satélite, la transmisión por cable no hace uso de un código interior, por lo que su potencia de corrección es algo menor. Es cierto que el canal usado es más controlado y puede asegurarse una mejor condición de recepción. Esto permite usar modulaciones QAM multinivel que son muy efectivas en cuanto a ancho de banda utilizado aunque más sensible a error que la QPSK para una misma relación Eb/No. En todo caso, las constelaciones de las modulaciones QAM multinivel se diseñan para minimizar la cantidad de bits erróneos en el caso de que puedan producirse errores. Para ello suelen usar ordenaciones basadas en codificación Gray.

Los tres pasos a realizar son:

-Mapeado de la señal de entrada a las señales IQ

-Modelado banda base de las señales IQ

-Modulación QAM de las señales IQ

MAPEADO DE LAS SEÑALES IQ

Los datos de entrada, que son la salida del intercalador convolucioneal, son una sucesión de bytes (en formato eléctrico paralelo) que deben pasarse a dos señales serie I,Q. Este proceso se realiza en tres pasos:

1.-Conversión byte a m_tupla

Según el nivel de la modulación (16,32 ó 64) los símbolos emitidos se corresponden con agrupaciones de 4,5 ó 6 bits (m_tupla; m=4,5 ó 6) y estas agrupaciones son las que se realizan en este paso. Se empieza por poner en serie la señal paralelo que llega, empezando por el MSB del byte. Se sigue tomando los bits por orden en grupos de 4,5 ó 6. Cada uno de estos grupos ya se denomina 'símbolo', pues dará lugar a un símbolo concreto.

2.-Codificación diferencial de los dos bits más significativos de la m_tupla.

No se trata de una codificación diferencial de los datos sino sólo de los dos bits más significativos de la m_tupla cuya misión es conseguir que la constelación sea

invariante a rotaciones π/2.

3.-Mapeado de la m_tupla resultante a señales multinivel IQ.

Este mapeado asigna a cada m_tupla resultante un valor a la señal I y otro a la señal Q para dar lugar en la modulación a un símbolo con una posición muy concreta en la constelación. Para conseguir esto, las señales IQ deben ser multinivel como se desprende del diagrama de constelación.

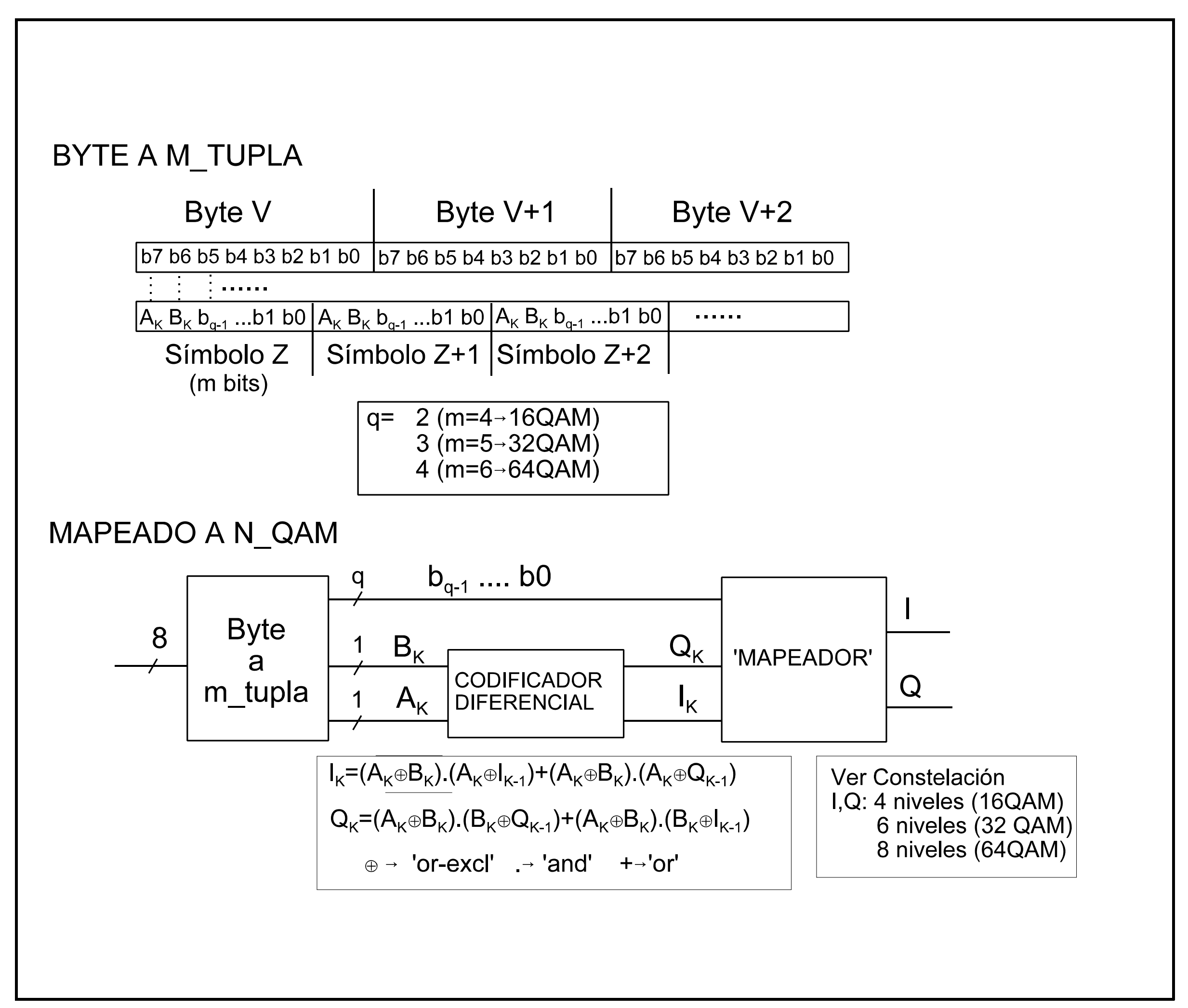


Fig.II-13.- Mapeado para modulación N_QAM

MODELADO BANDA BASE

Las señales multinivel IQ obtenidas anteriormente se filtran cada una con un filtro de respuesta coseno alzado que ya se ha descrito, pero con un parámetro de 'roll-off' menor que

en el caso de satélite. En cable es α=0.15. A este filtro se le denomina filtro de Nyquist.

Esto limita el régimen de símbolos, para un canal de 8 MHz de ancho de banda que son los que se usan en cable,a un valor máximo de 6.95Msímbolos/s.

MODULACIÓN QAM

Las señales IQ filtradas se introducen a un modulador de amplitud en cuadratura que modula cada señal en DBL con portadoras en fase y en cuadratura respectivamente. El resultado es una señal senoidal modulada cuya amplitud y fase en cada periodo de símbolo se corresponde con las de algún símbolo de la constelación.

Las constelaciones utilizables aparecen en la figura II-14.

Capacidad de transmisión

En la tabla siguiente aparecen varias posibilidades de capacidad de transmisión de señal TS_MPEG2 según la modulación usada para un ancho de banda ocupado del orden de 8MHz.

Capacidad de transmisión DVB-C				
Modulación	Ancho Banda	Symbol Rate en el cable	Bit rate de señal DVB	Bit rate de señal MPEG2
16QAM	7.86MHZ	6.84MBaudios	27.34Mbps	25.2Mbps
32QAM	7.96MHz	6.92MBaudios	34.61Mbps	31.9Mbps
64QAM	7.92MHz	6.89MBaudios	41.34Mbps	38.1Mbps

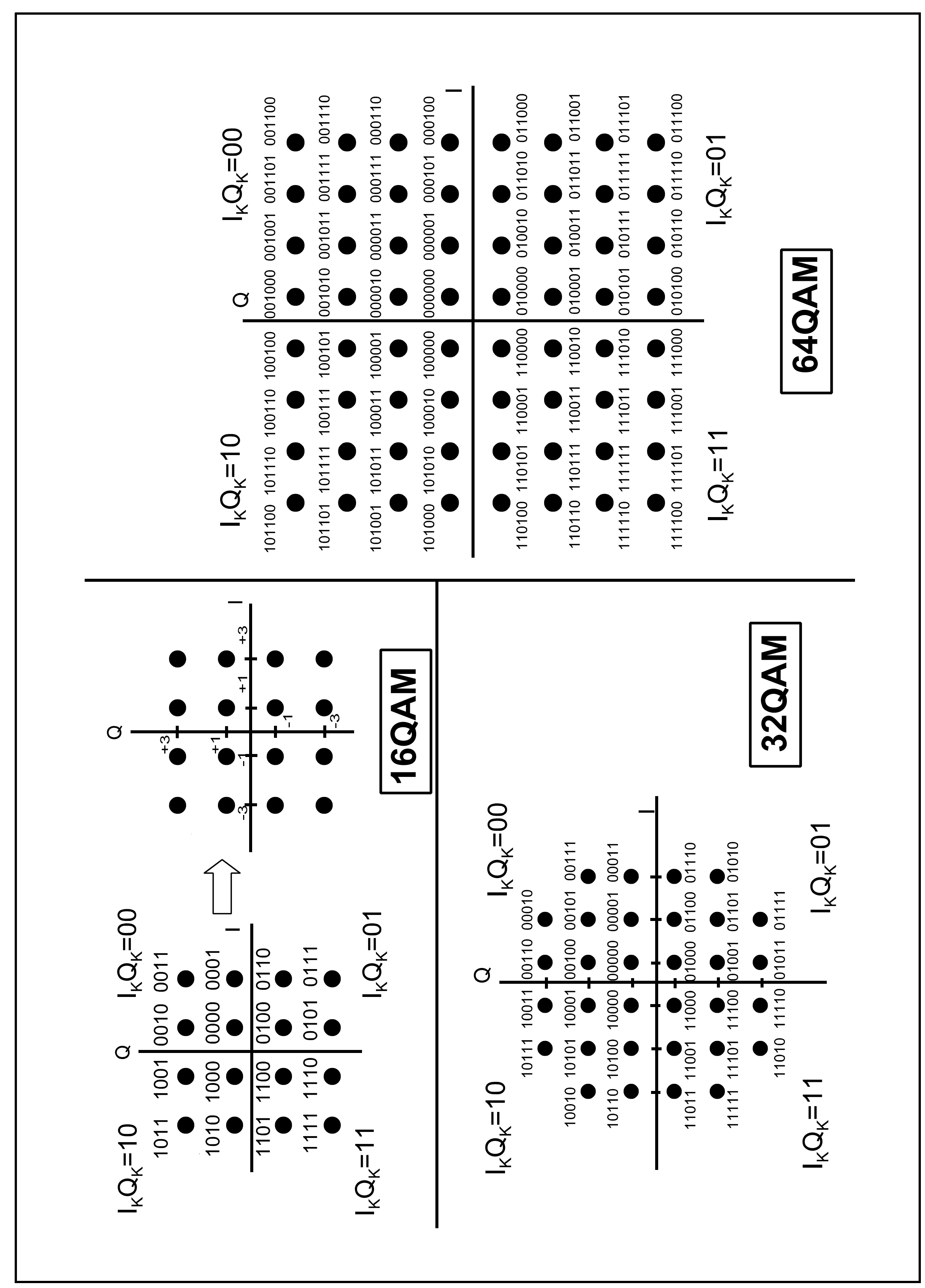


Fig.II-14.- Constelaciones para DVB-C

Filtrado de Nyquist: filtro de banda base

Como ya se ha indicado, las señales IQ son señales multinivel, pero al proceder de un procesado de bits, debe diferenciarse la apariencia matemática y la electrónica real de dichas señales. Como expresión de la información que continen (aspecto matemático) podría considerarse que dichas señales son 'deltas' de niveles discretos separadas el periodo del reloj de símbolo. Electrónicamente, sin embargo, los circuitos digitales trabajan con buses en los que los datos se corresponden con la existencia de un cierto nivel de tensión que se mantiene hasta que se transiciona al siguiente. La apariencia electrónica de las señales digitales es la de una señal analógica de pulsos. Si se analiza (muestrea) dicha señal en el punto medio de la duración de un símbolo se sigue identificando el nivel concreto y ,con él, el valor de la señal digital multinivel (es decir, se recuperan las 'deltas' de información).

Esta implementación 'analógica' de las señales digitales es la que justifica la existencia de interferencia entre símbolos debido a la deformación de las transiciones que llegan a modificar la amplitud de la señal en los puntos de análisis.

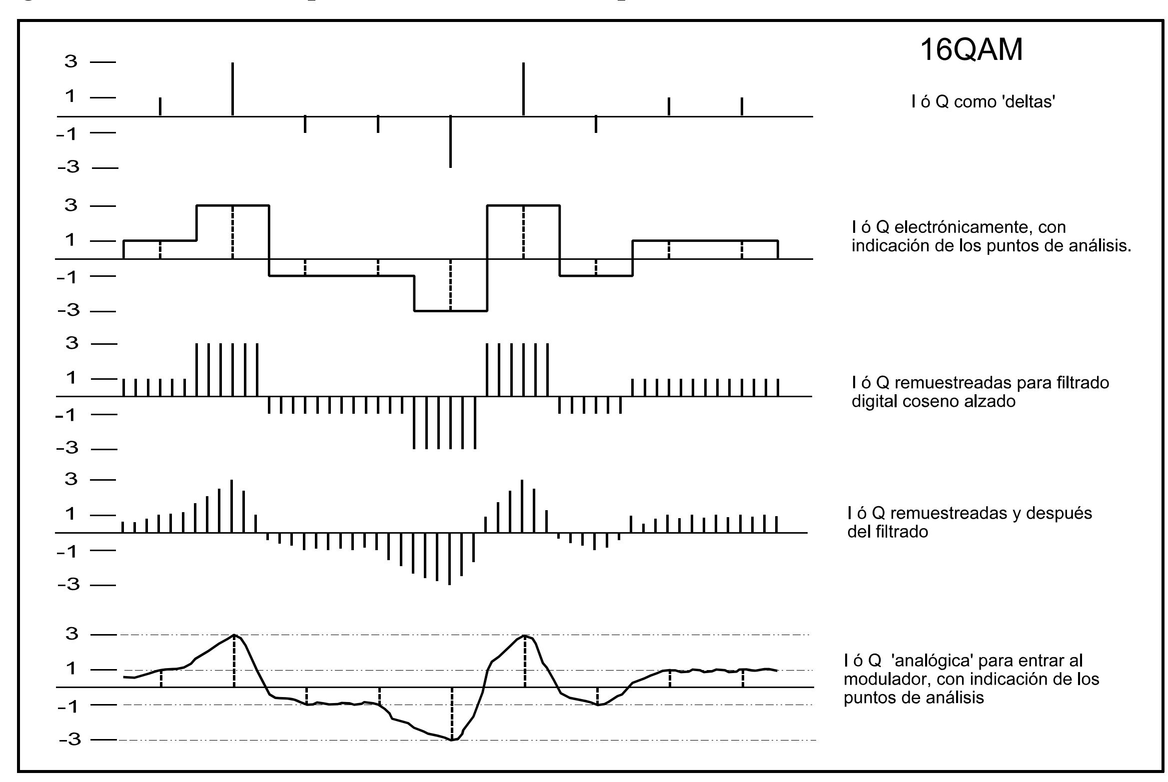


Fig.II-15.- Procesado banda base señal IQ-16QAM

El filtrado Nyquist de coseno alzado se realiza sobre esta señal electrónicamente analógica para 'moldear' sus transiciones según la respuesta impulsional del filtro y reducir (teóricamente eliminar) la interferencia de dichas transiciones sobre los puntos de análisis de la señal (ISI:'Inter Symbol Interference').

La implementación de filtros Nyquist analógicos para esta señal electrónicamente analógica está en desuso y se usan filtros digitales, pero para usarlos debe muestrearse la señal 'digital electrónicamente analógica' con una frecuencia mayor que la de símbolo para que el algoritmo pueda modificar sólo las transiciones.

El resultado a la salida del filtro es una señal discreta multinivel que se filtra paso bajo para convertir en analógica y atacar el modulador QAM. La señal que entra al modulador recibe el nombre de señal Banda Base. La modulación debe realizarse como si de modulación analógica se tratara para que tenga utilidad el filtrado de Nyquist en el codificador. No olvidar que en realidad se trata de señal digital aunque de apariencia electrónica analógica. Debido a todo este proceso, se suele decir que tiene lugar una conversión D/A, que pasa de bits a señal electrónicamente analógica.

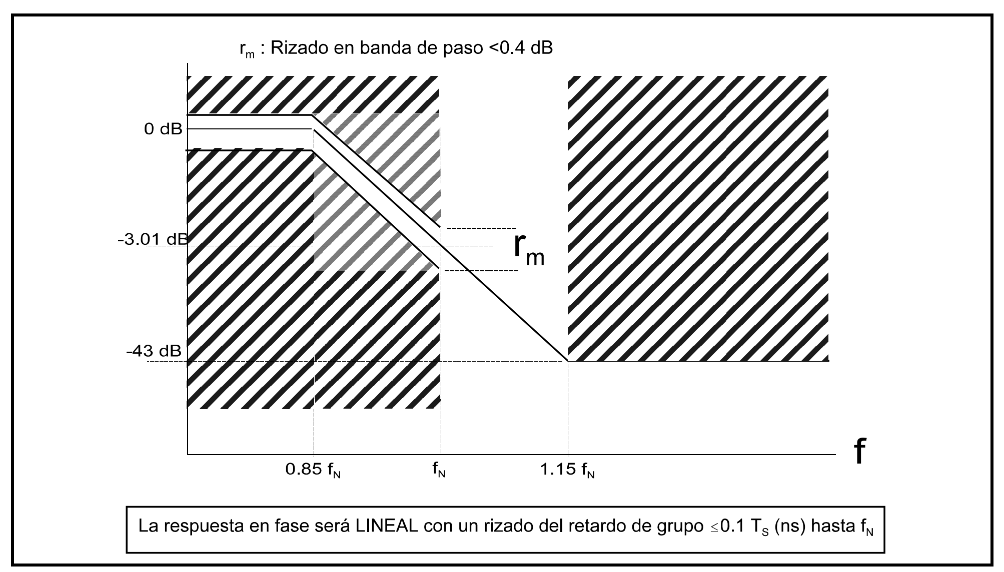


Fig.II-16.- Plantilla filtro coseno alzado

Teniendo en cuenta las limitaciones de los filtros digitales así como las limitaciones de otros procesos realizados en esta parte electrónicamente analógica (otros filtros, conversión D/A,...) se establece un patrón de respuesta del filtro de coseno alzado que admite unas variaciones respecto al valor teórico pero que no deben excederse (ver figura II-16).

DVB-TERRESTRE

Estructura de datos, código de canal y modulación

Como ya se indicó en cable, para asegurar la mayor compatibilidad posible y facilitar la transmidulación entre unas señales y otras, las normas DVB tienen en común los procesos:

-Adaptación del múltiplex TS y Aleatorización.

-Codificación exterior RS(204,188,t=8)

-Intercalado ('Interleave') Convolucional con profundidad I=11

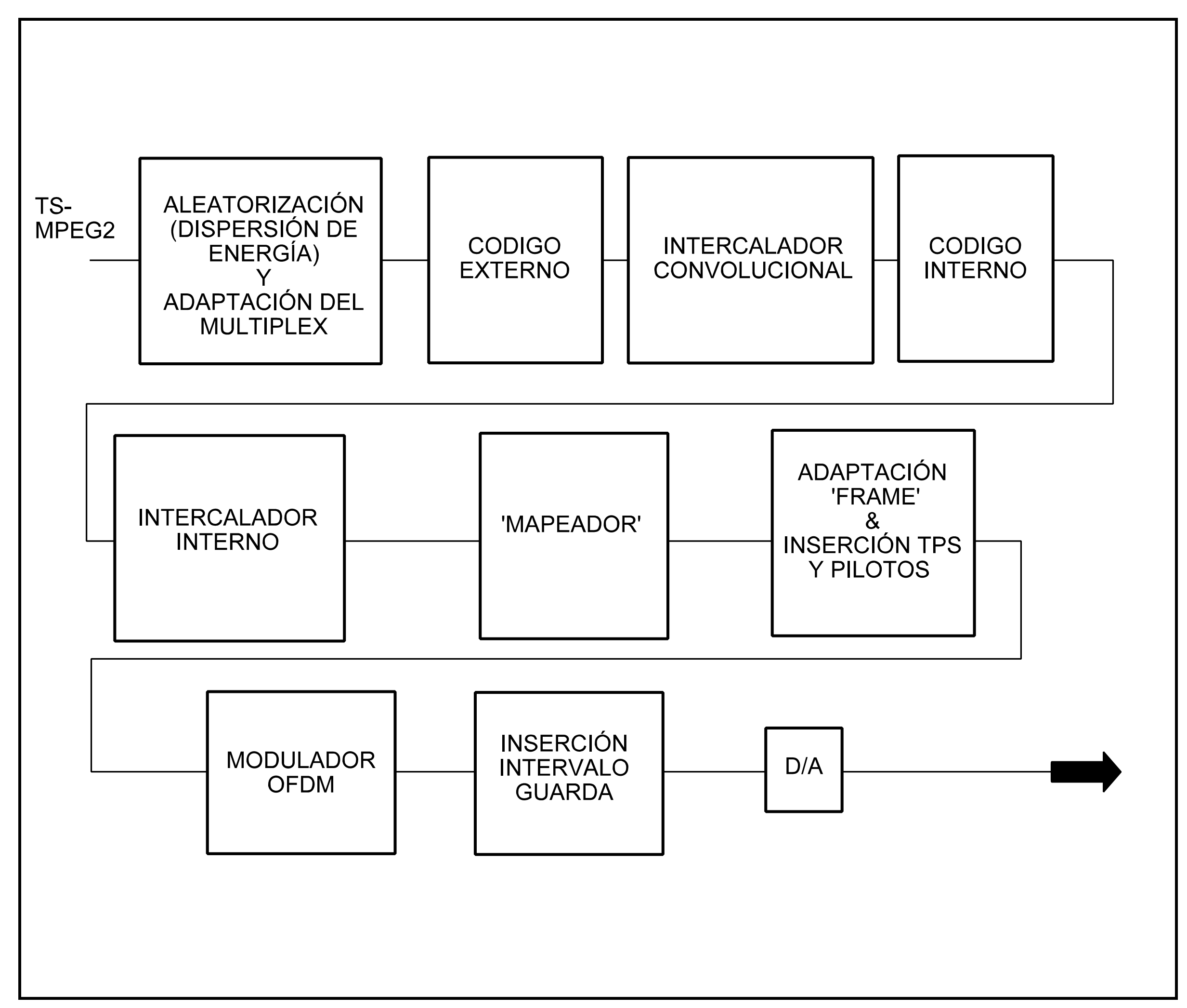


Fig.II-17.- Esquema general modulación DVB-T no jerárquica

En el caso de DVB-T, además se realiza una codificación interior tipo convolucional

y posteriormente se procede a una modulación de múltiples portadoras (OFDM) repartirendo los datos que salen del codificador interior entre ellas. Este proceso exige a su vez realizar un intercalado (no convolucional sino de bloque) de los bits entre los distintos símbolos IQ y también un intercalado de los símbolos entre las distintas portadoras (el llamado intercalado frecuencial). Las constelaciones a las que se mapean los símbolos en las portadoras pueden ser según se elijan: QPSK, 16QMA ó 64QAM.

Código interior convolucional

El código convolucional usado es el mismo que para satélite. La única diferencia de implementación es que la salida del 'puncturing' no aparece como dos señales IQ sino como una única señal serie. Esto se hace así para favorecer el posterior intercalado asociado a la modulación multiportadora.

Los parámetros del codificador son:

Codificador convolucional												
Codificador base			Relación del código ('rate')									
			1/2		**2/3**		**3/4**		**5/6**		**7/8**	
K	G_1 (X)	G_2 (Y)	Punct	d_{free}	Punct	d_{free}	Punct	d_{free}	Punct	d_{free}	Punct	d_{free}
7	171 oct.	133 oct.	X:1 Y:1	10	X:10 Y:11	6	X:101 Y:110	5	X:10101 Y:11010	4	X:1000101 Y:1111010	3
			$S=X_1Y_1$		$S=X_1Y_1Y_2$		$S=X_1Y_1Y_2X_3$		$S=X_1Y_1Y_2X_3Y_4X_5$		$S=X_1Y_1Y_2Y_3Y_4X_5Y_6X_7$	

Intercalador interno

Como puede verse en la figura II-18, la señal que sale del codificador convolucional 'punctured' pasa por un intercalador llamado interior que realiza dos intercalados como se va a explicar y la salida se 'mapea' a dos señales IQ dependiendo del tipo de modulación (constelación) elegido para las portadoras. Esas dos señales de salida modularán a las

portadoras OFDM. Cada pareja IQ (esto es, cada símbolo) modula una portadora, la siguiente pareja a la siguiente portadora y así hasta modular todas y volver a la primera.

No debe perderse de vista por tanto que las señales IQ son multinivel y cada pareja (I_k,Q_k) determina un símbolo o estado de la modulación de la portadora correspondiente 'k'. La entrada al mapeador es una sucesión de bits agrupados en m_tuplas (con m=2,4 ó 6 según la constelación elegida), cada m_tupla la denominaremos 'Y_n', y define un símbolo concreto en la constelación, que denominaremos 'z'. La entrada al intercalador interior es una sucesión de bits, que provienen del codificador interior y que denominaremos 's_i'.

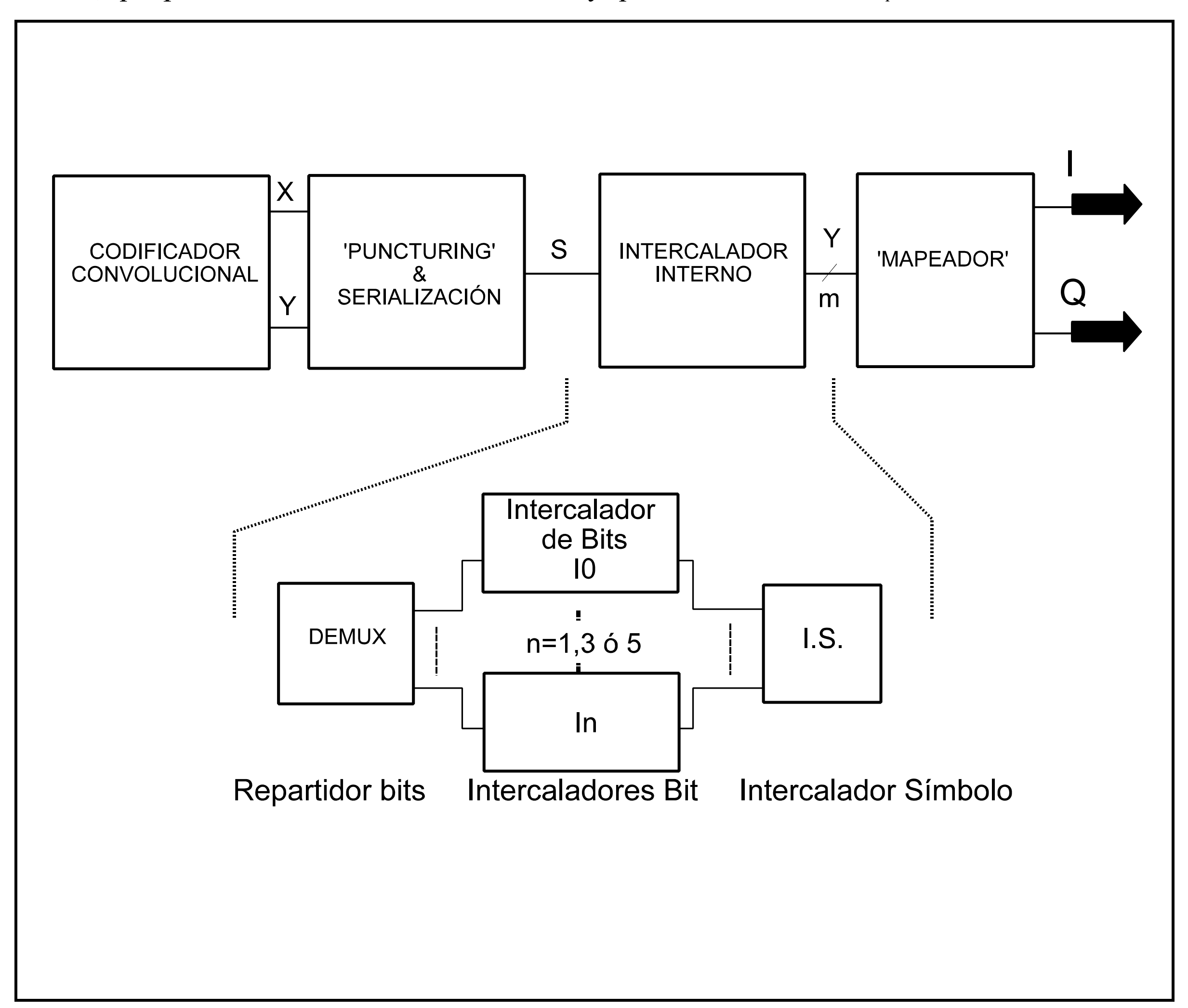


Fig.II-18.- Procesado 'interno'

Por concordancia con la nomenclatura de la norma, las salidas IQ se renombran como

Re[z] e Im[z] respectivamente, siendo 'z' la denominación genérica del símbolo de la modulación QPSK ó QAM.

Debe quedar claro al lector en este punto que en una modulación OFDM actual no existen realmente portadoras sinusoidales moduladas en cuadratura (módulo y fase) sino sólo los coeficientes que definen matemáticamente dicha modulación, que son Re[z] e Im[z], y que la existencia de dicha señal modulada se reduce a un número complejo 'z' que la define matemáticamente y que denominamos 'símbolo'. La obtención de un símbolo real OFDM, que sería la mezcla de múltiples portadoras moduladas, se realiza directamente, mediante técnicas de procesado digital, a partir de los valores complejos de los símbolos 'z' que representan a cada una de las modulaciones 'básicas'. No existen por tanto, físicamente, múltiples modulaciones de portadoras sinusoidales ortogonales en cuadratura. Sí existe la señal resultante que es el símbolo OFDM.

Intercalado interno de bit y de símbolo

La sucesión de bits 's_i' que entra al intercalador interno se separa en m chorros de bits (el valor de 'm' será acorde a la m_tupla que corresponda por la modulación) mediante un demultiplexador. En este demultiplexado ya se realiza cierta desordenación de bits. Cada chorro resultante pasa por un intercalador de bloque con una función de intercalado ('función de permutación') distinta, pero con un tamaño de bloque de intercalado que en todos los casos es de 126 bits. Las 'm' salidas que en cada momento entreguen los 'm' intercaladores (ver figuras II-19 y II-20: m_tupla [$a_{0,w}$ $a_{1,w}$ ...$a_{m-1,w}$] para el momento 'w') constituyen la m_tupla en formato paralelo (símbolo). Este símbolo está constituido por 'm' bits que no son consecutivos en la sucesión 's_j' de entrada,por lo que el símbolo contiene un intercalado de bits.

El símbolo sufre a continuación un intercalado con respecto a otros símbolos que van obteniendose, de modo que los símbolos Y_n que salen del intercalador de símbolos no mantienen el mismo orden con que entraron a él. Dado que cada símbolo que sale define la

modulación que se realiza a cada una de las portadoras OFDM, se dice que el intercalado de símbolos equivale a un intercalado de frecuencias portadoras, en el sentido de que dos portadoras consecutivas no llevan los datos de dos símbolos consecutivos. Este intercalado es de bloque y el tamaño del bloque es igual al número de portadoras OFDM, de modo que el intercalado de símbolos se hace dentro de un mismo símbolo OFDM. El bloque será de 1512 símbolos en el modo 2k y de 6048 símbolos en el modo 8k.

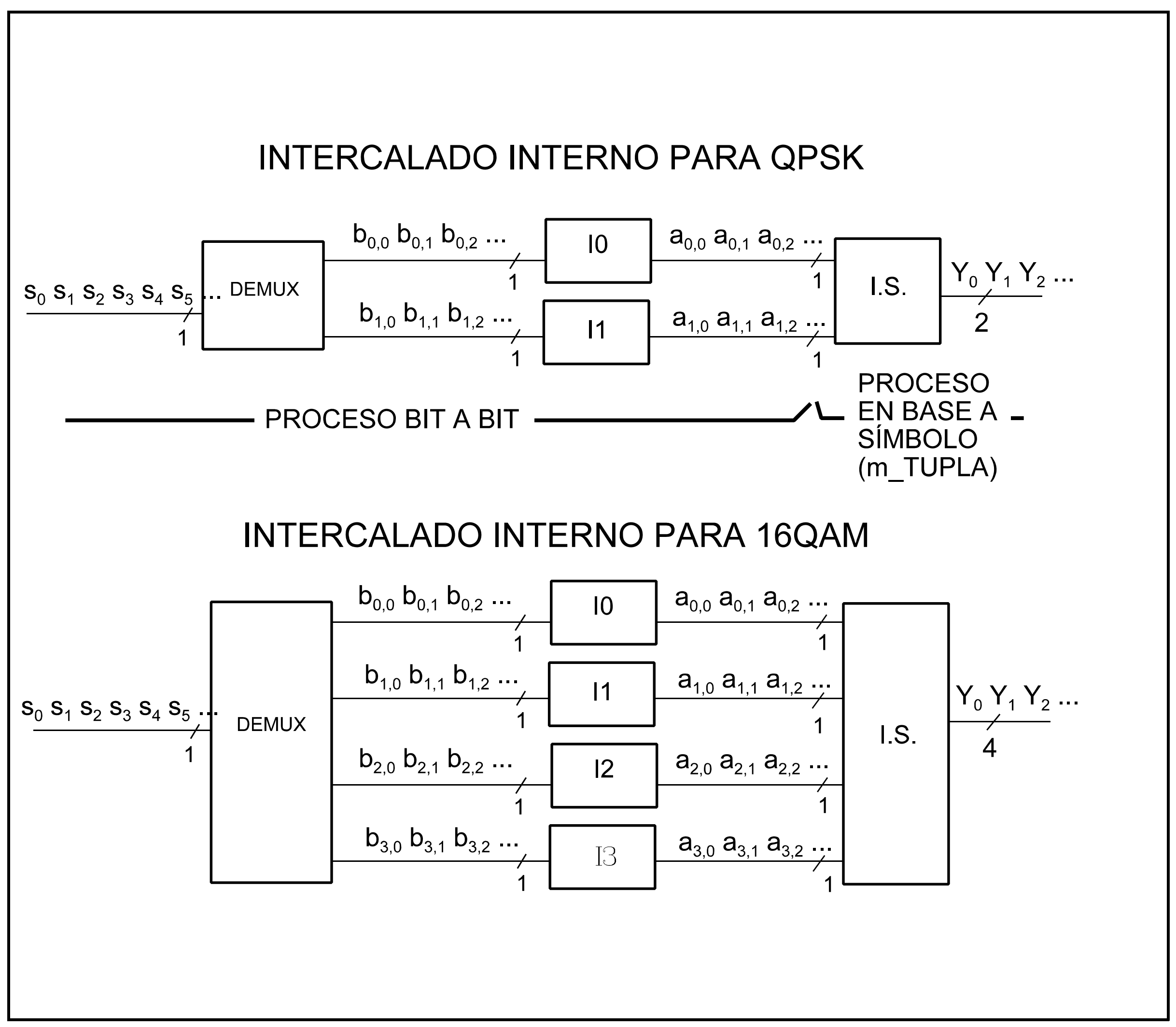


Fig.II-19.- Intercalado interno para QPSK y 16QAM

Las funciones que definen exactamente los procesos de demultiplexación, intercalado de bits e intercalado de símbolos son las siguientes:

DEMULTIPLEXADO PARA INTERCALADO INTERIOR DE BITS		
QPSK	16QAM	64QAM
$x_0 \rightarrow b_{0,0}$	$x_0 \rightarrow b_{0,0}$	$x_0 \rightarrow b_{0,0}$
$x_1 \rightarrow b_{1,0}$	$x_1 \rightarrow b_{2,0}$	$x_1 \rightarrow b_{2,0}$
	$x_2 \rightarrow b_{1,0}$	$x_2 \rightarrow b_{0,0}$
	$x_3 \rightarrow b_{3,0}$	$x_3 \rightarrow b_{0,0}$
		$x_4 \rightarrow b_{0,0}$
	$b_{chorro,orden}$	$x_5 \rightarrow b_{0,0}$

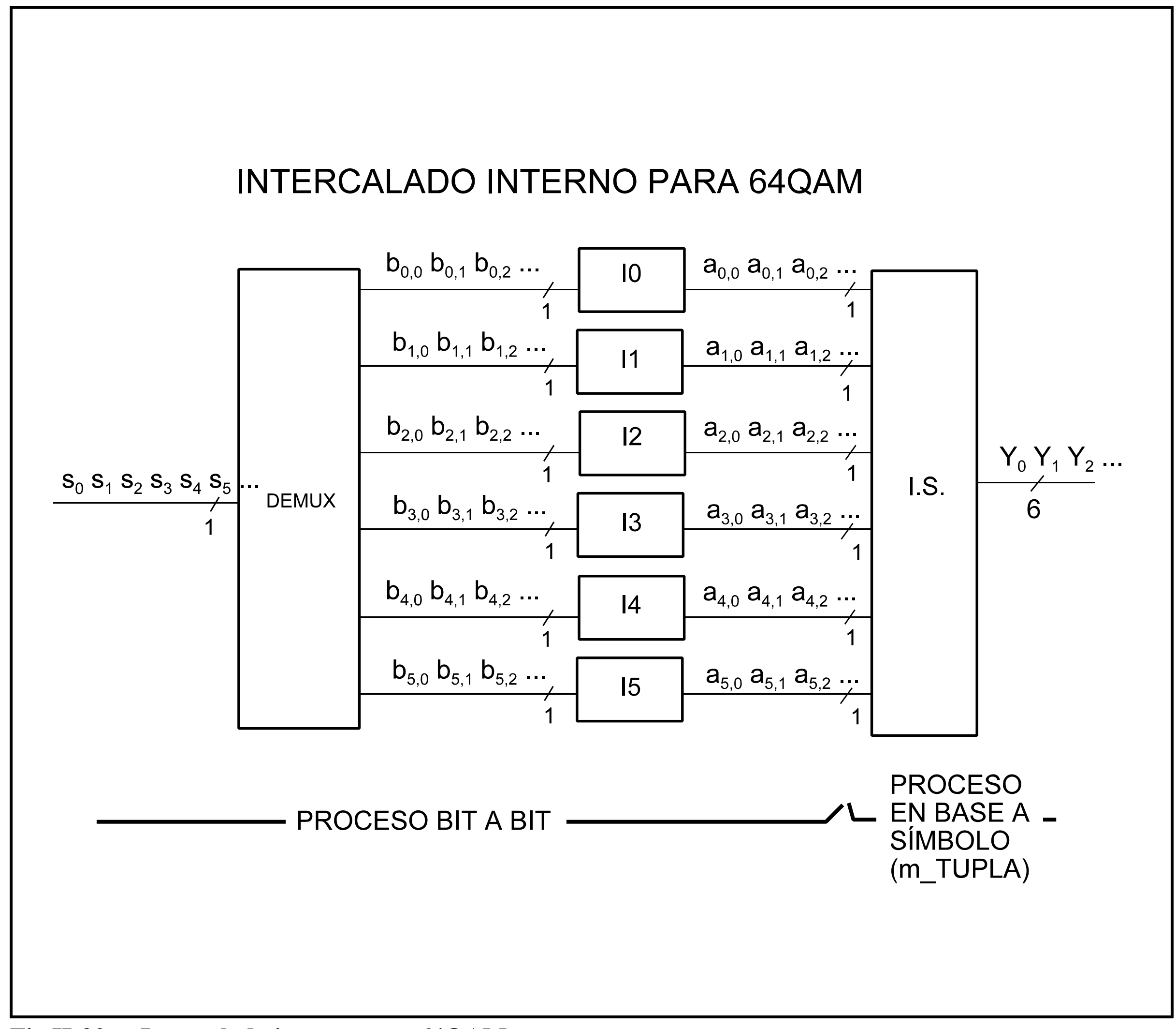


Fig.II-20.- Intercalado interno para 64QAM

INTERCALADORES DE BITS		
Bloque de entrada a intercalar (vector entrada)	$[b_{e,0}\ b_{e,1}\ b_{e,2}......\ b_{e,125}]$	126 bits en el intercalador 'e' (chorro 'e')
Bloque de salida intercalado (vector salida)	$[a_{e,0}\ a_{e,1}\ a_{e,2}......\ a_{e,125}]$	126 bits en el chorro 'e'
Función de intercalado	$a_{e,w}= b_{e,He(w)}$	Cada bit 'a' es un bit 'b' tomado en otro orden
Función de permutación 'He'	$H_0(w)=w$	Para I0 (chorro '0')
	$H_1(w)=(w+63) \bmod 126$	Para I1 (chorro '1')
	$H_2(w)=(w+105) \bmod 126$	Para I2 (chorro '2')
	$H_3(w)=(w+42) \bmod 126$	Para I3 (chorro '3')
	$H_4(w)=(w+21) \bmod 126$	Para I4 (chorro '4')
	$H_5(w)=(w+84) \bmod 126$	Para I5 (chorro '5')

Cada bloque o vector de salida se subdivide en m_tuplas formando símbolos.

INTERCALADOR DE SÍMBOLOS (Modo 8k)		
Bloque de símbolos de entrada (Vector de entrada)	$[Y_0'\ Y_1'\ Y_2'........Y_{6047}']$	Cada Y_w' se compone de 'm' bits $[a_{0,w}\ a_{1,w}....a_{m-1,w}]$ provenientes de los m chorros e=0,1....m-1.
Bloque de símbolos de salida (Vector de salida)	$[Y_0\ Y_1\ Y_2..........Y_{6047}]$	Cada Y_n tiene a su vez 'm' bits.
Función de Intercalado	Símbolo OFDM par: $Y_{H(q)}=Y_q'$ Símbolo OFDM impar: $Y_q=Y'_{H(q)}$	q=0,1,2,......,6047
Función de permutación H(q)	Ver tabla adicional	

<table>
<tr><th colspan="2">FUNCIÓN DE PERMUTACIÓN DEL INTERCALADOR DE SÍMBOLOS(8k)</th></tr>
<tr><td>Algoritmo función permutación</td><td>q=0
for(i=0;i<8192;i=i+1)
{
$H(q) = (i \bmod 2) \cdot 2^{Nr-1} + \sum_{j=o}^{Nr-2} R_i(j) \cdot 2^j$;
$if\ \ (H(q) < 6048q = q + 1;$
}</td></tr>
<tr><td>Vector R_i

(A partir de un vector R', permutando posición de bits)</td><td><table>
<tr><td>Bit R'</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Bit R</td><td>5</td><td>11</td><td>3</td><td>0</td><td>10</td><td>8</td><td>6</td><td>9</td><td>2</td><td>4</td><td>1</td><td>7</td></tr>
</table></td></tr>
<tr><td>Vector R'_i
(Cambia para cada 'i' del algoritmo)
Tiene 12 bits (en modo 8k)</td><td>R'_0=R'_1=[0 0 0.....0]
R'_2=[0 0 0 ..... 0 1]
Para i>2:
R'_i=R'_{i-1} anterior desplazado a la derecha 1 bit y calculado el más significativo $R'_1(11) = R'_{i-1}(0) \oplus R'_{i-1}(1) \oplus R'_{i-1}(4) \oplus R'_{i-1}(6)$</td></tr>
<tr><td colspan="2">Ver figura II-21</td></tr>
</table>

El bucle definido hace que para cada 'i' se obtenga un 'q' salvo para valores de H(q) superiores al límite, en cuyo caso 'i' se incrementa pero no se incrementa 'q'

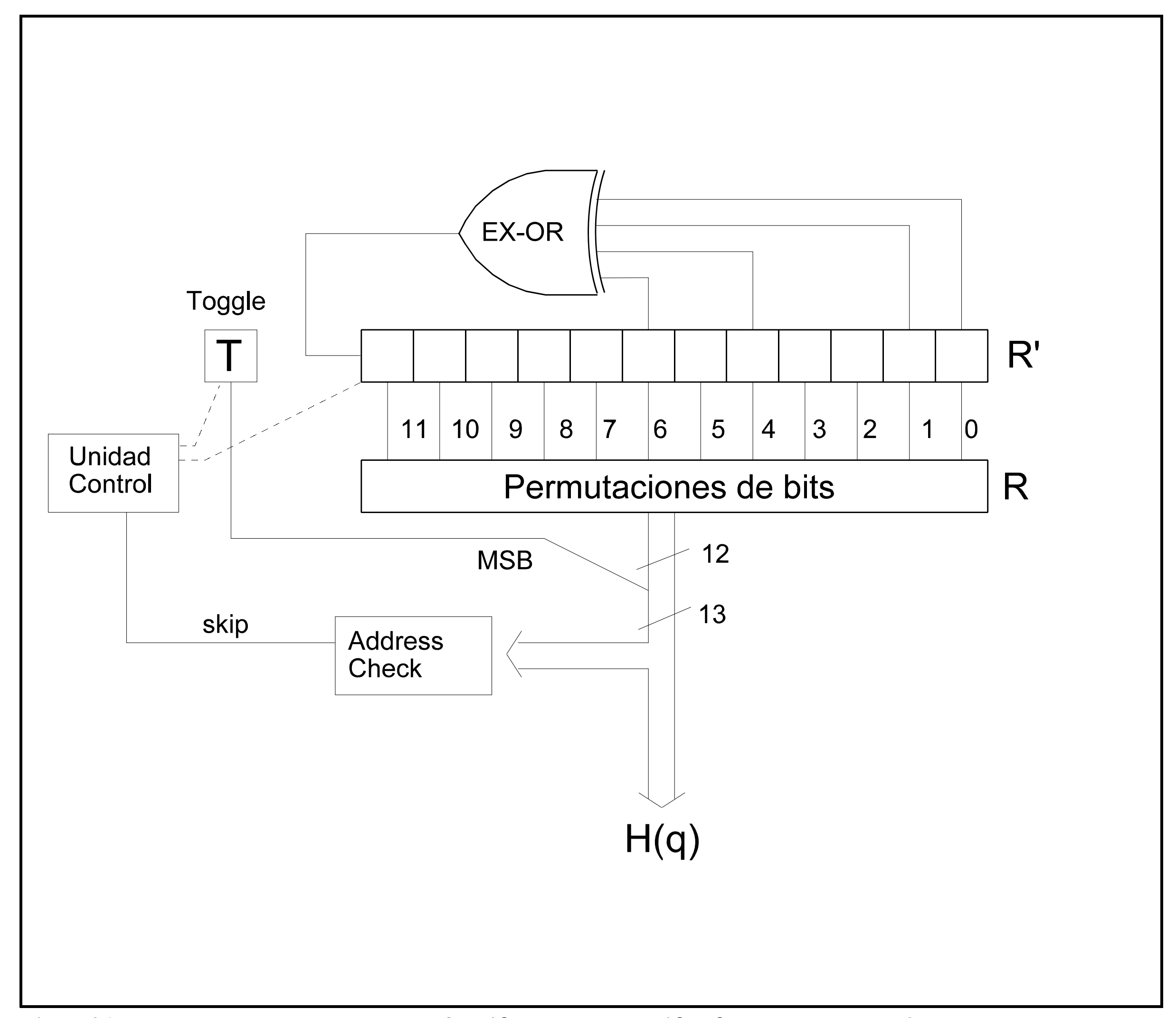


Fig.II-21.- Esquema del generador de función de permutación símbolos en modo 8k

Cuadro de datos

Este proceso se puede entender paso a paso (palabra a palabra), pero suele ser normal establecer caracterizaciones por conjuntos de datos que forman una estructura que se denomina 'cuadro de datos'.

La caracterización del proceso a nivel de cuadro de datos se realiza teniendo en cuenta que un bloque de entrada del intercalador de símbolos contiene 48 grupos de 126 palabras o m_tuplas (48x126=6048). El intercalador de bits daba a su salida m_tuplas

provenientes de 'm' bloques de 126 bits. La salida del intercalador de símbolos es de 6048 m_tuplas que se corresponde con todos los datos útiles que forman un símbolo OFDM. Como se verá existe una estructura más general que se denomina 'Frame OFDM' y que no debe confundirse con este 'cuadro de símbolo' que no es una estructura sino usa representación.

Mapeado

Aunque el proceso de intercalado interior, más concretamente el de símbolos, exige disponer de la casi totalidad de los 6048 símbolos de entrada para poder intercalarlos, la salida puede irse tomando en intervalos de tiempo menores, cada vez que tengamos un símbolo de 'm' bits definitivo. Cada uno de estos intervalos diremos que se corresponden con momentos de tiempo "n", de forma que el símbolo definitivo con todo el intercalado interior se caracteriza por un vector $Y_n =[y_{0,n}\ y_{1,n}....y_{m-1,n}]$ de 'm' bits.

A partir de este vector, debe obtenerse un símbolo de una modulación QPSK o QAM, con una componente en fase y otra en cuadratura como corresponda a la constelación. Para ello, el vector se divide en otros dos vectores por demultiplexado simple usandose los bits pares para constituir el vector que modula en fase o vector 'real' y los bits impares para el vector que modula en cuadratura o vector 'imaginario'. En la figura II-22 pueden verse las constelaciones así como la asignación del vector entrada a los dos vectores moduladores. El mapeado de bits a la constelación tiene en cuenta codificación Gray.

Como ya se ha comentado, no existe una modulación física de unas portadoras, por lo que no se usa la denominación IQ para estas señales sino Re[z] e Im[z]. Estas señales son multinivel tomando niveles según la combinación de bits de cada vector. Como se ve en la figura II-22, los niveles son ±1,±3,±5,±7. Como se verá más adelante estos valores son los coeficientes para 'sintetizar' el símbolo OFDM. Dado que un coeficiente se considera real y otro imaginario, hablamos de un dato complejo (llamado 'z') obtenido a partir de una m_tupla. Esto es importante para el procesado digital usado y también justifica que no se pueda usar una 32QAM pues 'm' sería impar y no podría separarse en dos vectores (real e imaginario) de igual longitud ambos.

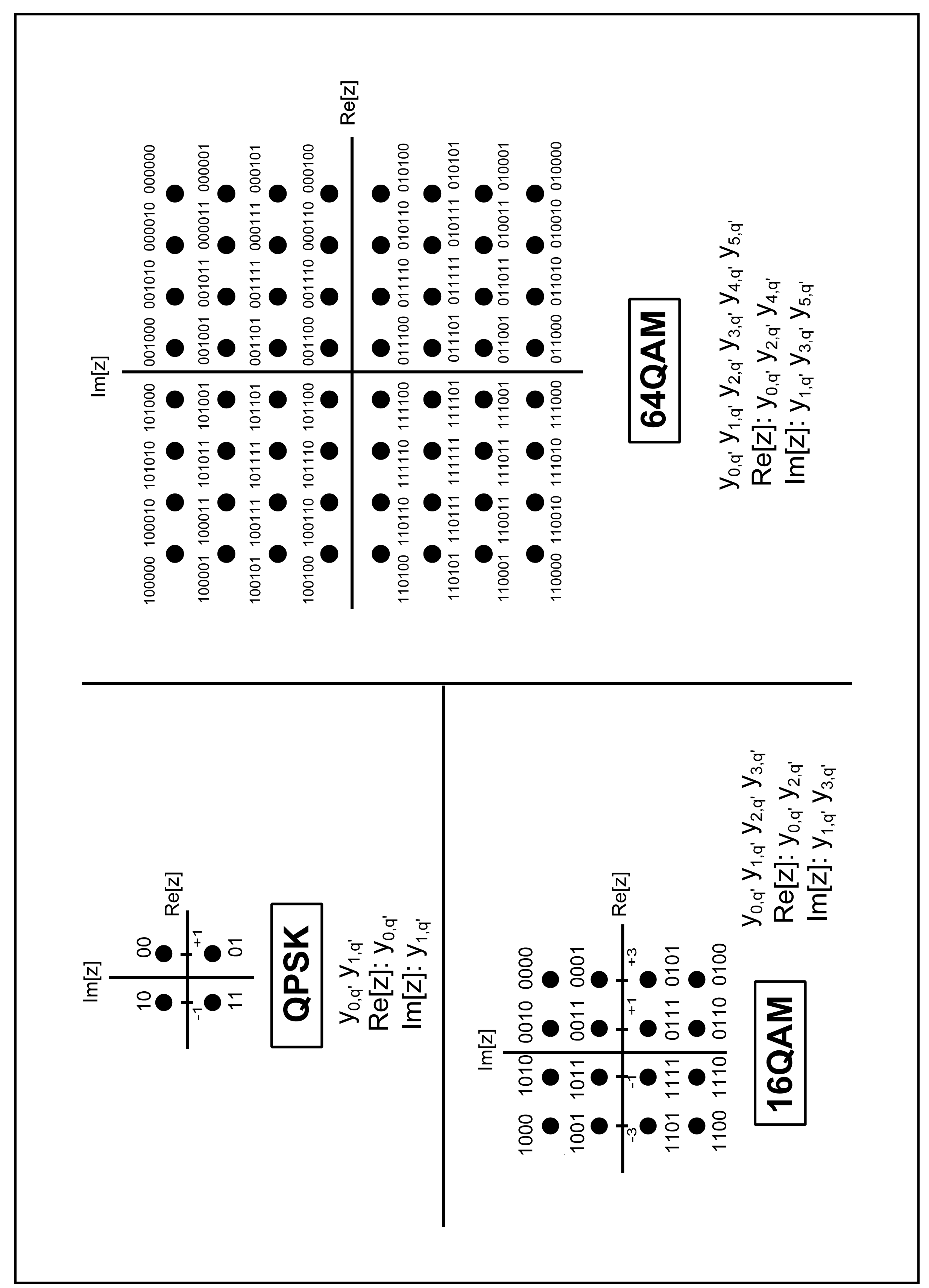


Fig.II-22.- Constelaciones DVB_T

Símbolo OFDM

Hasta este punto se ha explicado la obtención de los datos que definen las modulaciones de las 6048 portadoras que transportan señal 'útil'. Nos hemos ceñido a transmisión no jerárquica y en modo 8k principalmente.

La señal OFDM se configura como la transmisión secuencial de símbolos OFDM que se agrupan cada 68 símbolos en los denominados 'frames OFDM'. El significado de este agrupamiento se explicará un poco más adelante.

Cada uno de los símbolos OFDM no consiste exclusivamente en el resultado de la modulación de 6048 portadoras sino que, para facilitar aún más las posibilidades de la recepción de señal en condiciones diversas y en general adversas, se agregan portadoras con unos datos adicionales específicos (Portadoras piloto y portadoras TPS) y se alarga el símbolo mediante la repetición de la parte final de éste formando el denominado 'intervalo de guarda', que se situa al principio del símbolo.

El procesado digital que configura la señal OFDM se realiza en base a una frecuencia de reloj que es de 64/7 MHz, o sea, 9.14MHz , para utilizar canales de transmisión de ancho de banda de 8MHz. (Si el canal fuese a ser de 7 MHZ el reloj sería de 8MHz).

Las tablas siguientes muestran parámetros y duraciones de símbolos OFDM. Es interesante ver que la duración de la parte útil del símbolo es fija para cada modo (2k ó 8k).

Parámetros del símbolo OFDM		
Parámetro	Modo 8k	Modo 2k
Número de portadoras	6817	1705
Número de portadoras de datos	6048	1512
Numeración de las portadoras	0......6816	0......1704
Duración útil T_U	896μs	224μs
Intervalo de guarda Δ	A elegir (Δ/T_U=1/4,1/8,1/16,1/32)	A elegir (Δ/T_U=1/4,1/8,1/16,1/32)
Espaciado entre portadoras $1/T_U$	~1116 Hz	~4464 Hz
Espacio entre portadoras menor y mayor	~7.61MHz (~6.66MHz para canales de 7MHz)	~7.61MHz (~6.66MHz para canales de 7MHz)

Duraciones de símbolos OFDM (T es el ciclo del reloj básico: 1/9.14MHz=109.4ns)								
	Modo 8k				Modo 2k			
Intervalo de guarda Δ/T_U	1/4	1/8	1/16	1/32	1/4	1/8	1/16	1/32
Duración de parte útil T_U	896μs (8192 T ≡8k T)				224 μs (2048 T)			
Duración de intervalo de guarda Δ	224μs (2048T)	112μs (1024T)	56μs (512T)	28μs (256T)	56μs (512T)	28μs (256T)	14μs (128T)	7μs (64T)
Duración de símbolo $T_S=T_U+\Delta$	1120μs (10240T)	1008μs (9216T)	952μs (8704T)	924μs (8448T)	280μs (2560T)	252μs (2304T)	238μs (2176T)	231μs (2112T)

Duraciones del orden de 1ms para el modo 8k.

La señal OFDM, por tanto, se compone de símbolos que, en el modo 8k, contienen 6817 portadoras con datos de señal, de pilotos y de TPS, formando la parte útil con duración T_U que siempre dura 896μs. A esta parte útil se le antepone una repetición 'cíclica' de la parte final formando un intervalo de guarda Δ, teniendo el símbolo una duración total $T_S=T_U+\Delta$. El periodo básico de la señal es T=7/64 μs = 109.4 ns.

Como ya se ha indicado, los símbolos se agrupan en 'frames' de 68 símbolos. La señal OFDM puede por tanto expresarse matemáticamente con la siguiente fórmula que es la indicada por la norma:

$$s(t) = \mathrm{Re}\left\{ e^{j2\pi f_c t} \sum_{m=0}^{\infty} \sum_{l=0}^{63} \sum_{k=0}^{6816} C_{m,l,k} \cdot \Psi_{m,l,k}(t) \right\}$$

Las funciones Ψ(t) se analizarán en breve, pero puede adelantarse que junto con la exponencial compleja a frecuencia central representa a las 6817 portadoras. Los subíndices m,l,k de los sumatorios representan el número de orden del frame (m=0...∞), del símbolo OFDM dentro del frame (l=0...67) y de la portadora (k=0....6816).

Los coeficientes que 'modulan' cada portadora son valores complejos, con su parte real e imaginaria, que cumplen lo siguiente:

*Los 6048 valores de portadoras de datos:	Si modulación QPSK	$c = z / \sqrt{2}$
	Si modulación 16QAM	$c = z / \sqrt{10}$
	Si modulación 64QAM	$c = z / \sqrt{42}$
*Los pilotos y TPS	Independientemente de la modulación de datos	ver apartado correspondiente

El valor de 'z' ya se explicó anteriormente y lo que se hace es normalizarlos para que en todos los casos el valor de 'c' cumpla que $E[c \cdot c^*]=1$ (La esperanza o media de las potencias de los coeficientes para la constelación concreta se hace unidad). Los valores de 'c' para pilotos y TPS se indican en el apartado correspondiente y son valores siempre reales.

Función portadora Ψ(t)

Como ya se ha avanzado, la función portadora Ψ(t) junto con la exponencial compleja a frecuencia f_c forman la expresión matemática de cada una de las portadoras OFDM separadas adecuadamente la distancia $1/T_U$. Además, como puede apreciarse en la expresión siguiente, tomada de la norma, la ecuación sólo se define durante un intervalo de tiempo coincidente con la parte útil de cada símbolo OFDM.

$$\Psi_{m,l,k}(t) = e^{j2\pi \frac{k'}{T_U}(t-\Delta-l\cdot T_S-68\cdot m\cdot T_S)} \qquad para\ (l+68\cdot m)T_s \le t \le (l+68\cdot m+1)T_s$$

$$\Psi_{m,l,k}(t) = 0 \qquad otros\ t$$

Donde T_S es la duración de símbolo OFDM, T_U es la duración de la parte útil o inverso del espaciado entre portadoras, Δ es la duración del intervalo de guarda, m es el índice de orden del 'frame OFDM' que se trata, 'l' es el índice de orden del símbolo OFDM dentro del 'frame', 'k' es el índice de orden de la portadora a contar desde la primera y **k'** es el índice de cada portadora referida a la frecuencia central f_c del canal de transmisión OFDM. La relación entre k' y k es la siguiente: k'=k-3408 (para modo 8k).

Si nos ceñimos a un mismo símbolo OFDM, 'm' y 'l' tienen valor fijo y puede verse que el intervalo de validez de la función Ψ(t) es la duración del símbolo T_S, pero la función junto con la exponencial de f_c puede reescribirse en función de t (llamemos t_s para indicar que sus valores están en el intervalo de un símbolo concreto) pero con valores sólo durante la parte útil del símbolo T_U ($t_s-\Delta>0$), lo que es importante, pues sabemos que el comienzo del símbolo es el intervalo de guarda que consiste en repetición cíclica de una parte de T_U y no en parte calculada con coeficientes de datos específicos.

$$s(t_s) = \mathrm{Re}\left\{\sum_{k=0}^{6816} C_{m,l,k}\, e^{j2\pi\left[f_c t_s + \frac{k'}{T_U}(t_s - \Delta)\right]}\right\} \quad para\ \Delta \le t_s \le T_S$$

$$s(t_s) = \mathrm{Re}\left\{\sum_{k=0}^{6816} C_{m,l,k}\, e^{j2\pi\left[f_c + \frac{k'}{T_U}\right]t_s}\right\}$$

La última expresión recuerda a parte de la IDFT, lo que da lugar a que la mayoría de los moduladores OFDM se realicen sin osciladores ni moduladores reales sino directamente calculando la IDFT de los coeficientes de los datos que supuestamente iban a modular todas las portadoras.

'Frame' y ' SuperFrame' OFDM

Como ya se ha indicado, los símbolos OFDM se agrupan cada 68 para formar el denominado 'frame OFDM'. A su vez, cada 4 'frames' componen un 'superframe', numerándose los frames del 1 al 4.

'Transmission Parameter Signaling' (TPS)

Se trata de un conjunto de datos específicos sobre la transmisión que se incluyen en 68 portadoras de las 769 portadoras que en el modo 8k no son datos de señal original procesada. Los datos específicos que se transmiten como TPS aparecen en la tabla siguiente.

Resulta curioso comprobar que se trata de 68 datos, pero que el que se usen 68 portadoras en modo 8k es mera coincidencia, pues las 68 portadoras pertenecientes a un mismo símbolo OFDM llevan un único bit de los anteriores, repetido. Es un método de protección por si no se recibe bien alguna de las portadoras. Por tanto, para mandar los 68 bits son necesarios 68 símbolos consecutivos (es decir, 1 Frame). La situación es la misma en 8k que en 2k, la diferencia es que en 2k el número de portadoras (que repiten la información) en cada símbolo es menor que en 8k.

No debe confudirse el significado y valor en binario 'original' descrito en la tabla, de estos datos, con el valor del coeficiente $C_{m,l,k}$ usado para su modulación y transmisión. De momento sigamos analizando los datos en 'binario original'.

El bit de inicialización no corresponde propiamente a datos sobre la transmisión, y posteriormente se conocerá su uso, y por tanto no se incluye en la protección contra errores que se realiza con un código BCH que añade 14 bits a los 53bits de información trascendental. Se usa un BCH(67,53,t=2) que es un acortado del original BCH(127,113,t=2). El polinomio generador es:

$$h(x)=x^{14}+x^{9}+x^{8}+x^{6}+x^{5}+x^{4}+x^{2}+x+1$$

Como ya se sabe, los datos a codificar se rellenan con 60 ceros para usar el código original y una vez obtenido la palabra con redundancia, se eliminan para obtener la palabra de 67 bits.

Los datos s_{25}-s_{39} se refieren al superframe siguiente al usado en curso. Los restantes se refieren al superframe en curso.

1 bit		s_0	PRBS	Inicialización (No es fijo)
16 bits		s_1-s_{16}	35EE h(Frame 1 y 3) CA11 h(Frame 2 y 4)	Palabra de Sincronización
37 bits	23	s_{17}-s_{22}	'010111'	Longitud de datos que siguen (23 bits)
		s_{23},s_{24}	'00' Frame 1 '01' Frame 2 '10' Frame 3 '11' Frame 4	Número de Frame dentro del superframe actual
		s_{25},s_{26}	'00' QPSK '01' 16QAM '10' 64QAM '11' reservado	Constelación usada en superframe siguiente
		s_{27}-s_{29}	'000' No jerarquía '001' Jer. α=1 '010' Jer. α=2 '011' Jer. α=4 resto reservado	Información sobre jerarquía usada en superframe siguiente
		s_{30}-s_{32}	'000' 1/2 '001' 2/3 '010' 3/4 '011' 5/6 '100' 7/8 resto reservado	'Rate' del codigo convolucional de la señal o de la señal de alta prioridad si se usa jerarquía. Del superframe siguiente
		s_{33}-s_{35}	'000' 1/2 '001' 2/3 '010' 3/4 '011' 5/6 '100' 7/8 resto reservado	'Rate' del código convolucional de la señal de baja prioridad si se usa jerarquía o '000' si no se usa. Del superframe siguiente
		s_{36},s_{37}	'00' 1/32 '01' 1/16 '10' 1/8 '11' 1/4	Intervalo de guarda del superframe siguiente
		s_{38},s_{39}	'00' 2k '01' 8k resto reservado	Modo de transmisión del superframe siguiente
	14	s_{40}-s_{53}		Uso futuro
14 bits		s_{54}-s_{67}	Código BCH	Protección contra errores

Tabla de información TPS: 68 bits. El bit más a la izquierda se envía primero.

Modulación con los datos TPS

Cada dato TPS de los anteriores modula varias portadoras de un símbolo OFDM. Las portadoras usadas (llamadas portadoras TPS) no son cualesquiera sino las que se listan en la tabla siguiente

Portadoras TPS	
Modo 2k (17 portadoras)	Modo 8k (68 portadoras)
34, 50, 209, 346, 413, 569, 595, 688, 790, 901, 1073, 1219, 1262, 1286, 1469, 1594, 1687	34, 50, 209, 346, 413, 569, 595, 688, 790, 901, 1073, 1219, 1262, 1286, 1469, 1594, 1687, 1738, 1754, 1913, 2050, 2117, 2273, 2299, 2392, 2494, 2605, 2777, 2923, 2966, 2990, 3173, 3298, 3391, 3442, 3458, 3617, 3754, 3821, 3977, 4003, 4096, 4198, 4309, 4481, 4627, 4670, 4694, 4877, 5002, 5095, 5146, 5162, 5321, 5458, 5525, 5681, 5707, 5800, 5902, 6013, 6185, 6331, 6374, 6398,6581, 6706, 6799

Las 68 portadoras (en modo 8k) de un mismo símbolo llevan la misma información, correspondiente a un único bit de la tabla.

La modulación usada es DBPSK (PSK binaria diferencial) de modo que una misma portadora se modula en símbolos consecutivos con la diferencia respecto al bit anterior. La codificación diferencial se inicializa cada frame (68 símbolos) con un valor 'absoluto' enviado en el primer símbolo que no es otro que el bit de inicialización descrito en la tabla. Este bit no es fijo sino que se obtiene de un generador de secuencia pseudoaleatoria (PRBS) que se describirá más adelante al ser el mismo que el usado para las 'portadoras piloto'. Su valor se representa por 'w_k' y puede ser 0 ó 1. El valor del coeficiente para la modulación OFDM es: $C_{m,l,k} = (1-2w_k)+j0$

El resto de bits (marcados como s_i) darán la siguiente modulación:

Si s_i=0......$Re[C_{m,l,k}]=Re[C_{m,l-1,k}]$	Se repite el valor del símbolo PSK anterior
$Im[C_{m,l,k}]=0$	en todas las portadoras de este símbolo OFDM.
Si s_i=1......$Re[C_{m,l,k}]=-Re[C_{m,l-1,k}]$	Se invierte el valor del símbolo anterior.
$Im[C_{m,l,k}]=0$	

Sólo tienen parte real y su potencia está normalizada a 1 (Se dice que se emite con nivel de potencia 'normal' y cumple $E[C \cdot C^*]=1$). La constelación son dos puntos en el eje real con amplitudes ±1 (constelación PSK o BPSK como quiera decirse). Como ya se ha dicho esto sólo afecta a los valores de los coeficientes $C_{m,l,k}$, pues la modulación definitiva se obtiene mediante IDFT de todos los coeficientes.

Señales de referencia: Portadoras Piloto

Ya sabemos que en modo 8k, de las 6817 portadoras de un símbolo OFDM sólo 6048 llevan información de datos originales procesados. Otras 68 llevan los datos de transmisión TPS y las 701 portadoras restantes se usan para los pilotos de referencia.

La señal que se incluye en estas portadoras piloto no es de información concreta sino que corresponde a una secuencia pseudoaleatoria que en el receptor puede reconstruirse para comprobar las condiciones reales de la transmisión.

Secuencia Pseudoaleatoria de referencia

Se genera una secuencia contínua que obtiene un valor (bit) 'w_k' para cada portadora del símbolo OFDM desde la 0 hasta la 6816. Aquellos valores de 'k' que coincidan con portadoras de datos daran lugar a desechar el valor de w_k generado, pero aquellos valores de 'k' que correspondan a portadoras piloto darán lugar a la transmisión del valor w_k generado.

La secuencia se inicializa en cada símbolo OFDM de modo que su primer bit de salida coincide con la primera portadora (portadora 0). Es evidente que el generador se 'desprecia' durante el intervalo de guarda.

También hay que recordar que este generador se usa para el bit de inicialización de la modulación diferencial de los TPS, haciendo uso del primero de cada 68 símbolos OFDM.

El polinomio generador de la secuencia es: $x^{11}+x^2+1$. La inicialización se realiza poniendo todos los registros a '1'.

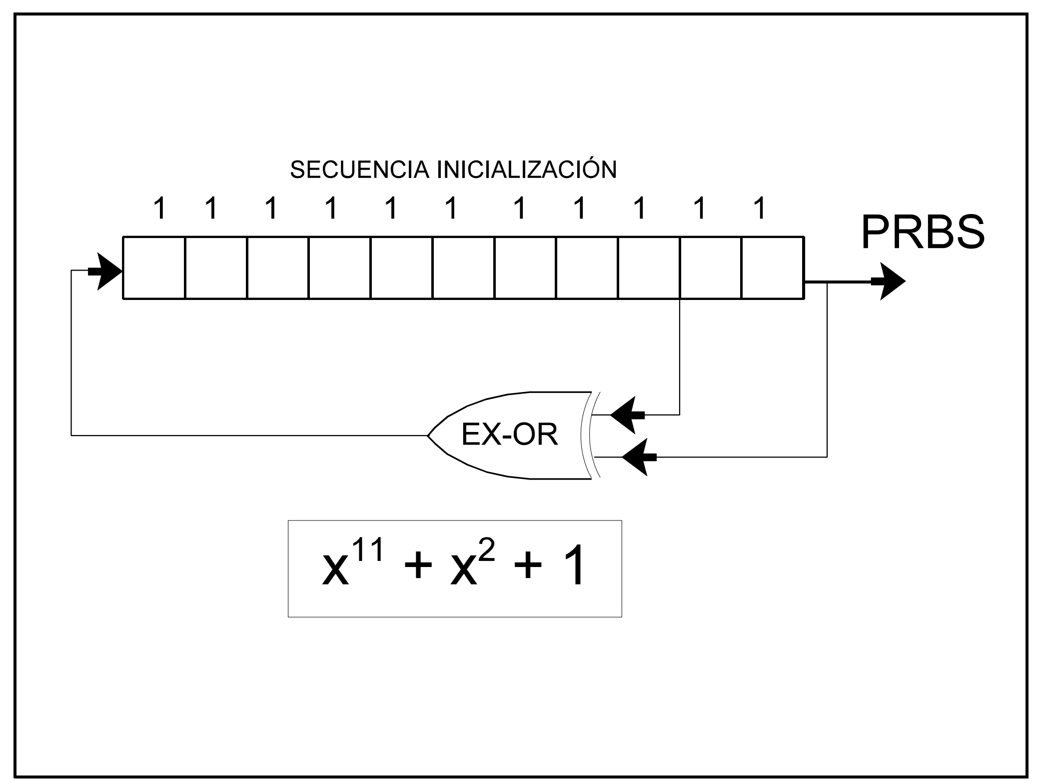


Fig.II-25.- Generador PRBS señales referencia w_k

Portadoras piloto

Se distinguen dos tipos de portadora piloto: pilotos contínuos y pilotos 'scattered'. Los contínuos aparecen en la misma portadora en todos los símbolos, mientras que los 'scattered' usan portadoras distintas en cada símbolo dentro de una secuencia de 4 símbolos (es decir, cada 4 símbolos se repiten las portadoras usadas).

Pilotos contínuos

En el modo 8k hay 177 portadoras asignadas a pilotos contínuos que se usan en todos los símbolos OFDM. La tabla siguiente las lista. Debe indicarse que algunas coinciden con portadoras para pilotos 'scattered' en según qué símbolos, lo cual no tiene importancia, pues lo esencial es que se envíe en esa portadora un valor de 'w_k'.

Portadoras para pilotos contínuos	
Modo 2k	**Modo 8k**
0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525 531 618 636 714 759 765 780 804 873 888 918 939 942 969 984 1050 1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491 1683 1704	0 48 54 87 141 156 192 201 255 279 282 333 432 450 483 525 531 618 636 714 759 765 780 804 873 888 918 939 942 969 984 1050 1101 1107 1110 1137 1140 1146 1206 1269 1323 1377 1491 1683 1704 1752 1758 1791 1845 1860 1896 1905 1959 1983 1986 2037 2136 2154 2187 2229 2235 2322 2340 2418 2463 2469 2484 2508 2577 2592 2622 2643 2646 2673 2688 2754 2805 2811 2814 2841 2844 2850 2910 2973 3027 3081 3195 3387 3408 3456 3462 3495 3549 3564 3600 3609 3663 3687 3690 3741 3840 3858 3891 3933 3939 4026 4044 4122 4167 4173 4188 4212 4281 4296 4326 4347 4350 4377 4392 4458 4509 4515 4518 4545 4548 4554 4614 4677 4731 4785 4899 5091 5112 5160 5166 5199 5253 5268 5304 5313 5367 5391 5394 5445 5544 5562 5595 5637 5643 5730 5748 5826 5871 5877 5892 5916 5985 6000 6030 6051 6054 6081 6096 6162 6213 6219 6222 6249 6252 6258 6318 6381 6435 6489 6603 6795 6816

Pilotos 'scattered'

La posición de las portadoras para pilotos 'scattered' varía de un símbolo a otro con una secuencia que se repite cada 4 símbolos y que se obtiene por la fórmula:

k=3 x (l mod 4) +12p , donde 'l' es el índice de orden del símbolo (0...67) y 'p' es un entero positivo que va tomando valores de uno en uno hasta completar los valores de 'k' en el margen de 0 a 6816 (en modo 8k). La operación 'x mod y' obtiene el resto entero de la

división x/y. Es evidente que no puede superar el valor de 3 en la fórmula. Algunos de los valores obtenidos con la fórmula coinciden con los pilotos contínuos como ya se ha dicho, concretamente con 43 pilotos contínuos en cada símbolo.

También puede comprobarse que la separación entre dos pilotos 'scattered' contiguos es de 12 portadoras (hay 11 portadoras entre dos pilotos 'scattered'), salvo respecto a la primera (0) y a la última (6816).

En la figura II-24 puede verse un ejemplo de la posición de pilotos 'scattered'.

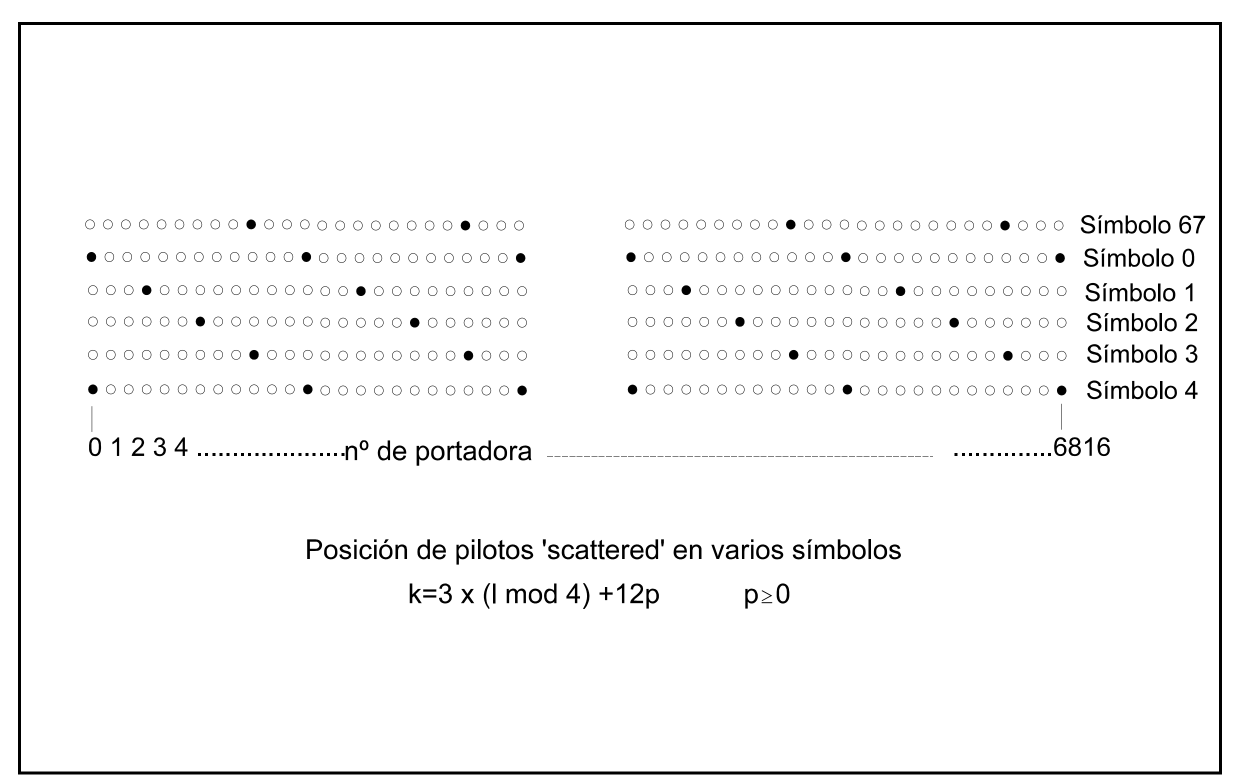


Fig.II-26.- Ejemplo de posición de pilotos 'scattered'

Visión general de las portadoras

Puede recontarse que en un símbolo OFDM, en modo 8k, hay 701 portadoras para pilotos, 68 para TPS y 6048 para datos, sumando un total de 6817 portadoras. De las 701 para pilotos, 177 son pilotos contínuos y hay 567 pilotos 'scattered' calculados, pero 43 de ellos

coinciden con contínuos ya contados.

También puede verse que resulta muy interesante para el estudio del sistema de modulación el que las portadoras pilotos, las portadoras TPS y las portadoras de datos se entremezclen en el conjunto de 6817 portadoras para formar el símbolo OFDM, lo que le da más inmunidad ante problemas en la respuesta espectral como atenuaciones o cancelaciones en zonas concretas del espectro (típico de escenarios de propagación multitrayecto o con redes de difusión de frecuencia única SFN o en distribuciones de cable con efectos de filtro peine por desadaptaciones de impedancia).

Modulación de portadoras piloto

La modulación se realiza también en BPSK pero no diferencial sino absoluta. Los niveles de la señal son mayores de los normales, no estando normalizados a la unidad. Los coeficientes de la modulación OFDM se calculan como:

$$C_{m,l,k} = [4/3 \cdot (1-2w_k)] + j0$$

Tienen sólo parte real (correspondiente a una BPSK) y su potencia cumple $E[C \cdot C^*] = 16/9$. La constelación es de dos puntos en el eje real tomando los valores ±4/3.

OFDM superframe y paquetes TS

Como ya se ha avanzado, un superframe está constituido por 4 frames. Como cada símbolo permite 6048 portadoras de datos procesados, cada frame tiene 68 símbolos y cada 4 frames se tiene un superframe, en un superframe se pueden transmitir 1645056 portadoras de datos.

Cada portadora de datos puede llevar 2,4 ó 6 bits (según la modulación usada) codificados convolucionalmente, a partir de menos bits (en una proporción marcada por el 'rate'), correspondientes a paquetes de 204 bytes protegidos RS que son desordenados por

intercalado.

Un aspecto importante de la estructura de frame y superframe es que un superframe siempre transmite un número entero de paquetes de 204bytes sea cual sea la modulación y el 'rate' usados. Dicho número de paquetes será distinto en cada caso, pero siempre entero, por lo que no se tiene que rellenar datos para cumplir un superframe ('stuffing bits') y todo superframe empieza siempre, en lo referente a información original transmitida, por un byte de sincro.

Paquetes de 204bytes transmitidos en un superframe para modo 8k			
'Rate' convolucional	QPSK	16QAM	64QAM
1/2	1008	2016	3024
2/3	1344	2688	4032
3/4	1512	3024	4536
5/6	1680	3360	5040
7/8	1764	3528	5292

Puede observarse la alta capacidad de datos de un sistema de modulación cuyos símbolos se transmiten a una bajísima frecuencia del orden de 1Ksímbolo/s ($T_S \sim 1$ms).

Capacidad de transmisión y ancho de banda

Profundizando en el aspecto de capacidad de transmisión de los datos binarios originales del TS_MPEG2 a la entrada del proceso DVB_T (antes del aleatorizador), la siguiente tabla da valores calculados (aproximados) de dichos valores para los distintos factores de la modulación seleccionables: 'rate' del código convolucional, constelación e intervalo de guarda. El cálculo se realiza para ajustar el ancho de banda de la señal OFDM a un canal de 8MHz. Este ancho de banda se ve afectado por la duración total del símbolo, incluido el intervalo de guarda.

Régimen Binario útil de señal TS_MPEG2 (Mbps aproximado)					
Modulación	'Rate' C.Convol.	Intervalo Guarda			
		1/4	1/8	1/16	1/32
QPSK	1/2	5.0	5.5	5.9	6.0
	2/3	6.6	7.4	7.8	8.0
	3/4	7.5	8.3	8.8	9.1
	5/6	8.3	9.2	9.8	10.1
	7/8	8.7	9.7	10.3	10.6
16QAM	1/2	10.0	11.1	11.7	12.1
	2/3	13.3	14.8	15.6	16.1
	3/4	15.0	16.6	17.6	18.1
	5/6	16.6	18.4	19.5	20.1
	7/8	17.4	19.4	20.5	21.1
64QAM	1/2	14.9	16.6	17.6	18.1
	2/3	19.9	22.1	23.4	24.1
	3/4	22.4	24.9	26.4	27.1
	5/6	24.9	27.6	29.3	30.1
	7/8	26.1	29.0	30.7	31.7

Debe recordarse que en lo referente a los parámetros descritos, la transmisión se

comporta más robusta ante errores de diverso tipo para constelaciones menos densas, mayor redundancia del 'rate' (mayor valor de este) y mayor intervalo de guarda. Como siempre, resulta una cuestión de compromiso elegir entre 'robustez' y capacidad de transmisión útil.

El espectro de la señal OFDM transmitida es prácticamente plano. Se obtiene por la superposición de todas las portadoras ortogonales equiespaciadas, las cuales tienen un módulo y fase cambiante símbolo a símbolo según los datos que las modulen. La densidad espectral de potencia de cada portadora a frecuencia f_k sigue una ecuación de $sinc^2$:

$$P_k(f) = \left[\frac{\sin[\pi(f - f_k)T_S]}{\pi(f - f_k)T_S} \right]^2$$

La suma de todas estas funciones para todas las portadoras dan un espectro prácticamente plano pero existe cierta ondulación del orden del espaciado entre portadoras. Debido a que la duración del símbolo T_S es mayor que el inverso del espaciado entre portadoras T_U,(por existir intervalo de guarda), el ancho del lóbulo principal de las sinc es más estrecho que 2 veces el espaciado entre portadoras, lo que justifica que el espectro resultante no sea plano sino con cierta ondulación. Esta ondulación depende del intervalo de guarda usado. Además, los extremos del espectro caen con la forma de la $sinc^2$ por lo que es bastante abrupto, siéndolo más en el caso de modo 8k que en el modo 2k al ser más largos los tiempos de símbolo. En ambos casos, el ancho ocupado es del orden de 7.6MHz. Existen unas ciertas 'colas espectrales fuera de banda' que suelen filtrarse.

Este espectro suele limitarse a canales de 8MHZ (±4MHz respecto a f_c). El valor de f_c suele tomarse de la fórmula $f_c=470 + 4 + 8\cdot i$ MHz salvo ajustes finos que puedan interesar. Esto hace que los canales sean los mismos que los usados en UHF para TV analógica que se corresponden con canales estandarizados. Cuando se usa un canal con señal digital adyacente a otro canal con señal analógica, debe asegurarse un filtrado adecuado para minimizar las interferencias de las colas fuera de banda del espectro digital. La norma especifica ciertos detalles de dichos filtrados.

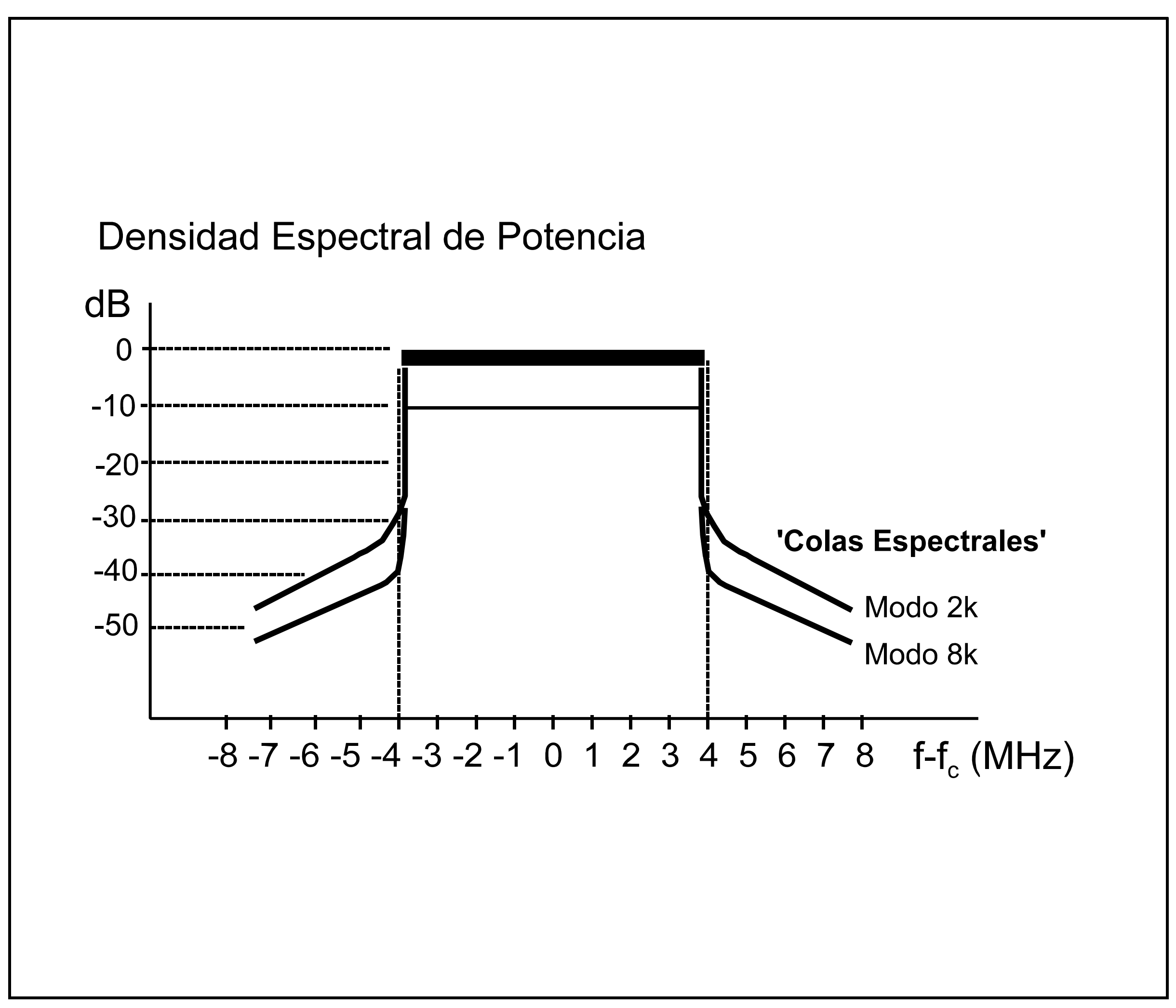


Fig.II-27.- Espectro típico DVB_T

Condiciones de recepción

La recepción adecuada de la señal DVB_T puede considerarse tal si la tasa de error obtenida después del procesador de corrección de errores de Reed-Solomon (a la entrada del decodificador MPEG-2) es del orden de $<10^{-11}$, es decir menor que un error sin corregir por hora. Esta es la situación denominada QEF ('Quasi Error Free'). Para ello, la tasa de error a la salida del decodificador de Viterbi (decodificador de códigos convolucionales) debe ser del orden de $2x10^{-4}$. Para asegurar esta condición debe existir una cierta relación C/N en la

recepción. La tabla siguiente muestra valores teóricos calculados según la norma. Estos valores se expresan para 3 condiciones de propagación:

-Sin multitrayecto, sólo ruido gausiano

-Con multitrayecto además de ruido gausiano:

*Canal RICEAN (Modelo de multitrayecto aplicable a receptores fijos)

*Canal RAYLEIGH (Modelo multitrayecto aplicable a receptores móviles)

		C/N (dB) requerida para BER=$2x10^{-4}$ a la salida de Viterbi		
Modulación	'Rate' convol.	Gausiano (Sin multitray)	Ricean (Rec.Fijo)	Rayleigh (Rec.Móvil)
QPSK	1/2	3.1	3.6	5.4
	2/3	4.9	5.7	8.4
	3/4	5.9	6.8	10.7
	5/6	6.9	8.0	13.1
	7/8	7.7	8.7	16.3
16QAM	1/2	8.8	9.6	11.2
	2/3	11.1	11.6	14.2
	3/4	12.5	13.0	16.7
	5/6	13.5	14.4	19.3
	7/8	13.9	15.0	22.8
64QAM	1/2	14.4	14.7	16.0
	2/3	16.5	17.1	19.3
	3/4	18.0	18.6	21.7
	5/6	19.3	20.0	25.3
	7/8	20.1	21.0	27.9

Recuerdese que cada una de estas combinaciones de parámetros permitía una cierta capacidad de datos. Además, estos valores no deben usarse de forma crítica en instalaciones, sino que debe añadirse algún margen de seguridad así como el 'margen de implementación'

del receptor en concreto que se use.

Uso de la IFFT para realizar la modulación OFDM

Ya se ha indicado la relación que existe entre la fórmula de la señal OFDM y la de la IDFT cuando se particulariza para un símbolo OFDM concreto y el intervalo de tiempo usado es el de T_U.

Esto invita a usar dicha operación para implementar un modulador OFDM. Como se sabe, dicha operación matemática se realiza por procesado digital mediante un algoritmo de resolución rápida denominado IFFT. En el receptor se propone usar la FFT para la demodulación.

Este proceso trabaja en banda base (no considera las portadoras a su valor real de frecuencia alrededor de la f_c correspondiente) y obtiene las N muestras 'x_n' correspondientes a la parte útil T_U de cada simbolo, mediante una N-IFFT. A continuación, pero fuera del proceso IFFT se repiten cierta cantidad de los últimos valores obtenidos, situándolos al comienzo de los datos para obtener el intervalo de guarda. Este proceso se repite símbolo a símbolo obteniéndose una representación compleja en banda base de la señal OFDM. Posteriormente, un proceso de cambio de frecuencia obtendrá la señal real s(t) centrada en la zona requerida del espectro.

Sin embargo, debe prestarse atención en la conversión de los valores $C_{m,l,k}$ a coeficientes de entrada de la IFFT (X_q), pues puede haber problemas.

Observese la similitud de ecuaciones, pero también los puntos de posible conflicto, para, por ejemplo, el símbolo 0 del frame 0 (m=0, l=0) en modo 8k.

$$s(t) = \mathrm{Re}\left\{ e^{j2\pi f_c t} \sum_{k=0}^{6816} C_{0,0,k} \cdot e^{j2\pi \frac{k'(t-\Delta)}{T_U}} \right\} \quad con\ k' = k - 3408$$

$$x_n = \sum_{q=0}^{N-1} \frac{X_q}{N} e^{\frac{j2\pi nq}{N}}$$

Existen 6817 valores de C, pero N valores de X_q (N es potencia de 2).

Elección de la frecuencia central en 'banda base'

Como ya hemos indicado este es un cálculo en banda base. Debe hacerse pensando en reconstruir una señal que se define modulada con una frecuencia central.

La función Ψ para la portadora central (i.e. 3408 en modo 8k) es siempre constante y de valor unidad en cada símbolo pues k'=0. Esta portadora se emitirá a frecuencia f_c y no presentará nunca discontinuidades de fase cuando se module con el mismo valor en todos los símbolos. Existen ,en resumen, dos alternativas para la asignación:

1.-Asignar la portadora central al coeficiente central q=N/2, lo que equivale a asignar la portadora central al término de frecuencia mitad de la de muestreo (máxima frecuencia). (Recuerdese que los N puntos de la FFT se corresponden con frecuencias equiespaciadas entre 0 y $f_{muestreo}$ (0 y 2π))

2.-Asignar la portadora central al término q=0, esto es tomar la portadora central como término de componente contínua (DC) y considerar que el espectro varía entre $-f_m$ y f_m (-π y +π).

En cualquiera de los dos casos, deben rellenarse de ceros los extremos hasta completar los 8192 (8k) valores para el cálculo de la IFFT. Este relleno no afecta al resultado salvo en que tendremos 8192 valores de salida (T_U=8192 T) en vez de 6817. Estos 8192 valores forman la misma forma de onda del símbolo OFDM que la que formaría la IFFT de los 6817 coeficientes si se pudiera implementar, salvo por un factor de escala proporcional a N.

Otro aspecto que queda por analizar es la conversión de esta señal compleja banda base a una señal real centrada en una frecuencia portadora, típicamente una frecuencia intermedia para la modulación ya que posteriormente se heterodina para obtener la portadora exacta en el canal UHF de uso.

Este proceso se corresponde con la parte de la fórmula del símbolo en la que aparece la exponencial compleja en f_c y la expresión de "Parte Real: Re{ }". Dado que la IFFT obtiene valores complejos, que denominaremos 'd_n', se tiene que:

$$s(t) = \mathrm{Re}\left\{e^{j\omega_c t} \cdot d_n\right\} = d_{nr}\cos(\omega_c t) - d_{ni}\sin(\omega_c t)$$

Por lo que la obtención del símbolo a la frecuencia f_c exige una modulación QAM con la parte real e imaginaria de los datos resultantes de la IFFT invirtiendo la fase de la modulación cuadratura. Esto último significa que las modulaciones de cada portadora (0...6816) quedan realmente moduladas con inversión del eje imaginario respecto a las constelaciones definidas para el mapeado. Esto no es problema pues en la demodulación se invierte este efecto.

Debido a los aspectos anteriores, debe cuidarse de cumplir que:

1.-El espectro de la señal emitida no quede invertido respecto al previsto en las especificaciones y que por tanto, la portadora de mayor frecuencia emitida se corresponda con la modulación con el coeficiente $C_{m,l,6816}$.

2.-Tampoco tendrá la señal emitida un eje imaginario invertido comparado con el especificado. Esto ocurre si en la expresión de la señal emitida s(t) se sustituyen los coeficientes $C_{m,l,k}$ por sus complejos conjugados.

Las causas típicas de los dos errores anteriores suelen ser:

-Usar una conversión incorrecta para pasar de complejo a real. Provoca inversión de espectro y de eje imaginario.

-Usar el algoritmo de FFT en vez del de IFFT. Provoca inversión del eje imaginario.

-Usar circuitos de heterodinación, para subir frecuencias, que inviertan el espectro debido al oscilador local usado.Provoca también inversión del eje imaginario. Si el proceso se repite en un segundo paso todo queda correcto (up-conversion en dos etapas).

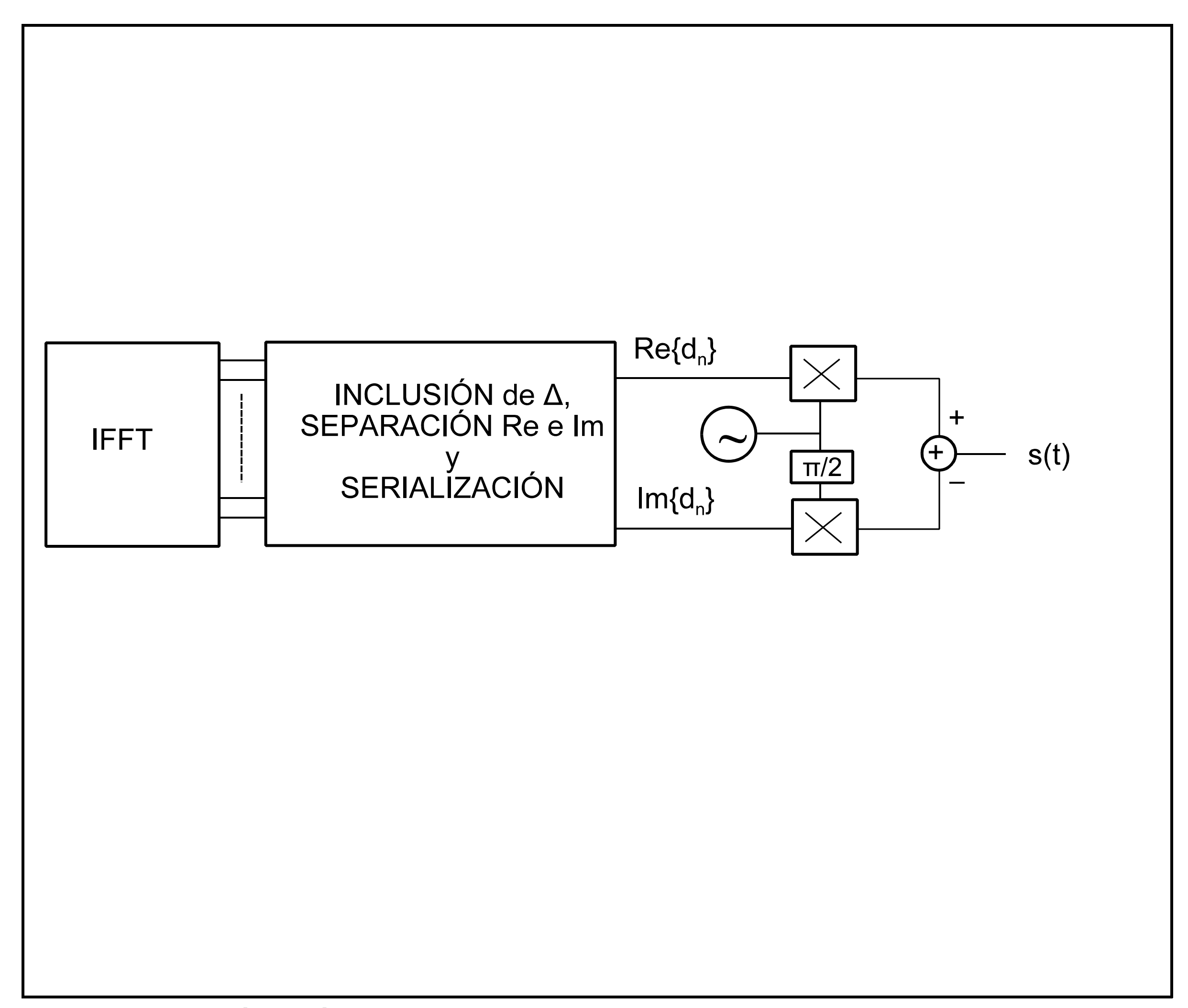


Fig.II-28.- Obtención del símbolo s(t) a partir de los datos de la IFFT.

LISTADO DE FIGURAS

BIBLIOGRAFÍA

Capítulo I

Information technology: Generic coding of moving pictures and associated audio information. Partes 1 (sistemas) y 2 (vídeo).
Draft International Standard ISO/IEC 13818.1994.

Architecture and VLSi implementation of the MPEG2:MP@ML video decoding process. Stojanic and Ngai. SMPTE Journal. Febrero 1995

Characterization of the recordable digital videodisc for recording MPEG-2 data stream using EFM channel code. Isailovic. SMPTE Journal. Septiembre 1995

Matching MPEG-2 to applications. Lookabough. SMPTE Journal. Enero 1995

MPEG overview. Baron and Wilson. SMPTE Journal. Junio 1994

Switching facilities in MPEG-2: necessary but not sufficient. Weiss. SMPTE Journal. Diciembre 1995

MPEG video: a simple introduction. S.R.Ely. EBU Technical Review nº266, p.12 Winter 1996

Capítulo II

Digital broadcasting systems for television, sound and data services; Framing structure, channel coding and modulation for 11/12 GHz satellite services. European Telecommunication Standard ETS 300 421. E.T.S.I. Diciembre 1994

Digital broadcasting systems for television, sound and data services; Framing structure, channel coding and modulation for cable systems. European Telecommunication Standard ETS 300 429. E.T.S.I. Diciembre 1994

Digital video broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television. European Standard (Telecommunications series) EN 300 744. E.T.S.I. 1997.

Experimental and simulation results on an OFDM modem for TV broadcasters. M.Saito et al. SMPTE Journal. Enero 1996

1996 Television progress report. R.B.Kisor. SMPTE Journal. Enero 1997

Practical implementation of digital television. W.Y.Zou and J.A.Kutzner. SMPTE Journal. Abril 1997

A prototype VLSI solution for digital terrestrial TV receivers conforming to the DBV-T standard. F.Scalise et al. SMPTE Journal. Noviembre 1997.